SPECIALE BASIC

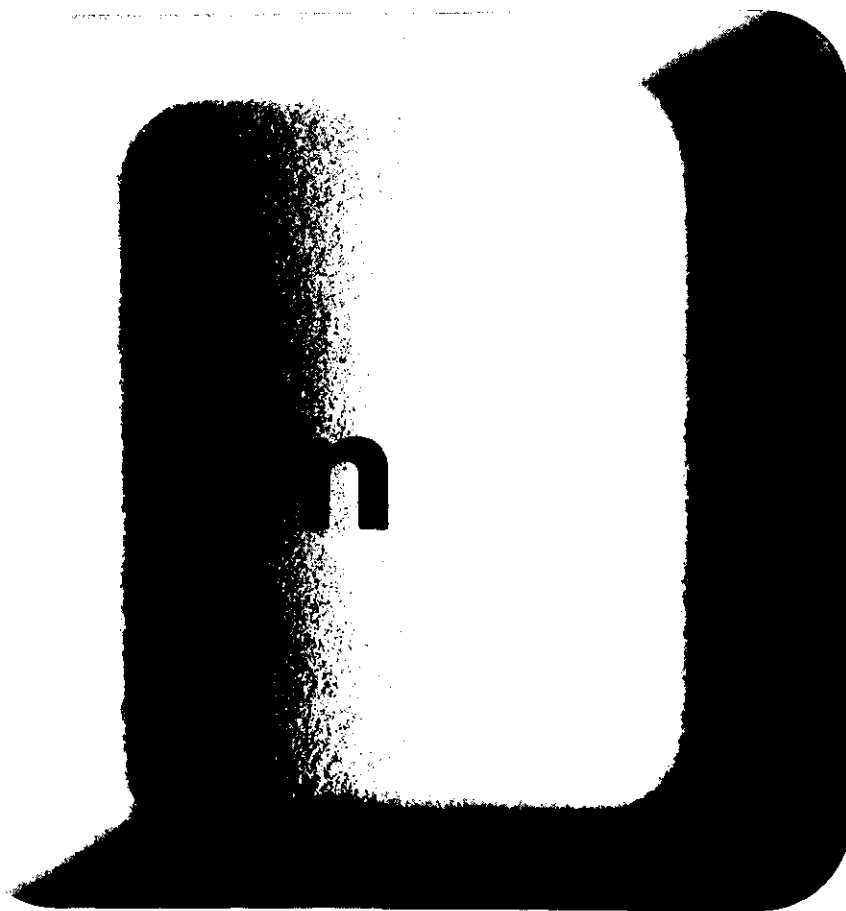
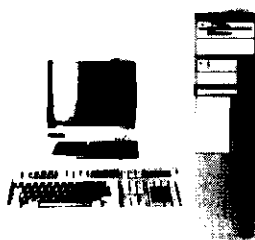
ANNO 3 - N. 13
GIUGNO 1990
L. 14.000
Frs. 21.00

IL MENSILE JACKSON PER GLI UTENTI DI AMIGA

- CONTIENE DEMO-VERSION DI DRAGONS BREATH
- Professional Draw
- Flicker Fixer
- Acquisition 1.3
- CeBIT
- MIDI Editor/Librarian
- I segreti del Workbench
- Quattro espansioni a confronto

INSERTO LE PAGINE DI Amiga Transactor per Amiga
EDIZIONE ITALIANA

Spedizione in abbonamento postale Gruppo III/70 Taxe percue (Tassa riscossa) Milano Ferr. Corr.

GRUPPO EDITORIALE JACKSON

DA COSA
NASCE COSA, NASCE COSA, NASCE COSA.

Da oggi, grazie ai PC Commodore, problemi tecnici e problemi pratici si risolvono più facilmente. Commodore Italiana, infatti, ha creato e garantisce in prima persona una linea di personal capace di rispondere alle esigenze di tutti e di lavorare e dialogare con tutti: dall'utente più sofisticato al neofita più acerbo. Da oggi, invece di scegliere un semplice PC, scegliete di fare un investimento garantito da Commodore Italiana.

Commodore

PC COMMODORE. FACILE IL DIFFICILE.

AMIGA MAGAZINE
ANNO 3 - NUMERO 13
GIUGNO 1990

Sommario

13

30

Direttore Responsabile: Paolo Reina
Coordinamento Tecnico e Redazionale: Massimiliano Anticoli
Tel. 02/6948287
Redazione: Stefano Albarelli (On Disk), Romano Tenca (Transactor)
Segreteria di redazione: Elena Ferré - Tel. 02/6948254
Art Director: Marcello Longhini
Copertina, Grafica e Impaginazione elettronica: Cristina Turra
Collaboratori: Lucio Bragagnolo, Cesare Palmieri, Liliana Simonetti, Stefano Riva, Sergio Ruocco, Sebastiano Vigna, Daniele Cassanelli, Mauro Ballocchi, Aldo e Andrea Laus, Marco Tortolina, Gianni Biagini, Renato Acciardi, Stefano Paganini

Group Publisher: Pierantonio Palerma
Direzione Coordinamento Operativo: Graziella Falaguasta
Publisher Area Consumer: Filippo Canavese
Pubblicità: Filippo Canavese
Direzione Sviluppo Pubblicità: Walter Bussolera

SEDE LEGALE
Via P. Mascagni, 14 - 20122 Milano

DIREZIONE - REDAZIONE
Via Pola, 9 - 20124 Milano - Tel.: 02/69481
Fax: 02/6948238 Telex 316213 REINA I

PUBBLICITA'
Via Pola, 9 - 20124 Milano - Tel.: 02/6948254
ROMA - LAZIO E CENTRO SUD Via Lago di Tana, 16
00199 Roma
Tel.:06/8380547 - Fax: 06/8380637

INTERNATIONAL MARKETING
Tel.:02/6948233

DIREZIONE AMMINISTRATIVA
Via Rosellini, 12 - 20124 Milano Tel.: 02/69481
Fax: 02/6948238

UFFICIO ABBONAMENTI
Via Rosellini, 12 - 20124 Milano - Fax: 02/6948489
Telex 333436GEJ IT - Tel.: 02/6948490 (nei giorni di martedì, mercoledì, giovedì, 14.30 - 17.30)

Prezzo della rivista: L. 14.000 prezzo arretrato L. 28.000
Abbonamento annuo Italia L. 124.000, Estero L. 248.000
I versamenti vanno indirizzati a:
Gruppo Editoriale Jackson SpA
Via Rosellini, 12 - 20124 Milano, mediante l'emissione di assegno bancario o per contanti. L'abbonamento può essere sottoscritto anche utilizzando il c/c postale 11666203

CONSOCIATE ESTERE
GEJ Publishing Group Inc. Los Altos Hills
27910 Roble Blanco
94022 California - Tel.: (001-415-9492028)
Grupo Editorial Jackson - Calle Alcantara, 57
280016 Madrid - Tel.: 1/4017365

Stampa: F.B.M. (Gorgonzola)
Fotolito: Foligraph (Milano)
Distribuzione: Sodip - Via Zuretti, 25 - 20125 Milano

Il Gruppo Editoriale Jackson è iscritto al Registro Nazionale della stampa al N. 117 Vol. 2 foglio 129 in data 17/8/1982.
Spedizione in abbonamento postale gruppo III/70
Aut.Trib. di Milano n. 102 del 22/2/1988

Il Gruppo Editoriale Jackson possiede per "Amiga Magazine" i diritti di traduzione di **Compute!'s Amiga Resource, Compute! Publications Inc. e Transactor for the Amiga, Transactor U.K.**
Amiga Magazine è una rivista indipendente non connessa alla Commodore Business Machine Inc., né con la Commodore Italiana S.p.a.- C64 e Amiga sono marchi registrati dalla Commodore Business Machine.

Associato al

Mensile associato all'USPI
Unione Stampa
Periodica Italiana

Consorzio
Stampa
Specializzata
Tecnica

Testata aderente al C.S.S.T. non soggetta a certificazione obbligatoria per la presenza pubblicitaria inferiore al 10%

Il Gruppo Editoriale Jackson pubblica anche le seguenti riviste:

Amiga Magazine Games - Pc Games - Guida Videogiochi
Supercommodore 64 e 128 - Pc Software - Fare Elettronica
Bit - Computer Grafica & Desktop Publishing - Informatica Oggi
Informatica Oggi Settimanale - NTE Compuscuola
Pc Floppy - Pc Magazine - Automazione Oggi
Trasmissioni Dati e Telecomunicazioni - Elettronica Oggi
EO News settimanale - Strumenti Musicali - Watt
Meccanica Oggi - Strumentazione e Misure Oggi

Editoriale

Tante novità

Anche questo mese Amiga Magazine è "pieno" di novità.

Lo speciale del mese è dedicato al BASIC, il linguaggio più conosciuto e più utilizzato nell'universo informatico !

Per l'attualità, ovvero "Amiga Magazine in giro per il mondo", vi presentiamo le novità del CeBIT di Hannover, dell'European Computer Trade Show di Londra, e della Newtronic di Genova.

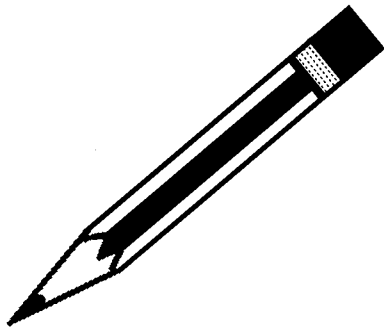
Nella parte hardware questo mese parliamo di: Flicker Fixer, contro lo sfarfallio dello schermo, di PAL-RGB Converter e di Flashfire, un rivoluzionario joystick-mouse!!

Per il software ci soffermiamo sull'ultima versione di Acquisition e sul potentissimo Professional Draw.

Inoltre, il nostro tecnico mette a confronto quattro espansioni di memoria; continuiamo il nostro viaggio nei segreti del MIDI e per finire abbiamo intervistato i migliori produttori di software ludico americani.

Per questo mese ci congediamo e... non dimenticatevi che c'è anche l'inserto di TRANSACTOR, dedicato a tutti i programmatori, e ON DISK con il favoloso disco dove abbiamo incluso, grazie un accordo fra Amiga Magazine e SoftMail, la demo-version del favoloso Dragons Breath !!

Arrivederci in edicola a Luglio con un numero ancora pieno di sorprese (Amiga 3000 e 2.0, ex 1.4, bastano?).

La redazione

Pascal e Amiga

Spett. redazione,
sono un possessore di un fantastico Amiga 2000. A scuola, sugli Olivetti M24 del laboratorio di Informatica, ho visto girare il Turbo Pascal della Borland. Vorrei sapere se esiste una specie di Turbo Pascal per Amiga.
Ringrazio anticipatamente per la risposta.

Domenico Beraldi

No, purtroppo non è stata ancora sviluppata una versione del Pascal con un ambiente integrato come il Turbo Pascal. Il Pascal più diffuso per Amiga è l'MMC Pascal della Metacomco, che però è molto lontano dal Turbo Pascal (Edia Borland), sia per l'interfaccia grafica (inesistente nell'MMC Pascal) che, soprattutto, per la velocità.
Tuttavia esiste un linguaggio simile, diretto discendente del Pascal di cui avrai sicuramente sentito parlare: si tratta del Modula-2 con il quale la programmazione risulta addirittura più semplice che in Pascal. Di questo linguaggio di programmazione esistono tre pacchetti con un ambiente integrato come il Turbo Pascal, e sono: M2Sprint, prodotto da M2S, M2-Amiga (Interface Technologies) e Benchmark Modula-2 (AvantGarde).

Precisazioni

Invitiamo tutti i nostri lettori a non inviarci francobolli.

La redazione

Il fantasma LFormat

Spett. Redazione,
posseggo da pochi mesi un Amiga 500, che ho cambiato con il mio glorioso C64. Ho iniziato subito ad utilizzare il CLI per "sfruttare" al meglio la macchina. Ho una curiosità da soddisfare: non ho capito la funzione dell'opzione LFORMAT presente in più di un comando AmigaDOS 1.3. Distinti saluti.

Matteo Nobili

La funzione dell'opzione LFORMAT è semplice: esso permette di formattare l'output, ossia di ottenere un risultato di un comando in una determinata forma. Ad esempio: il comando List è uno di quei comandi che incorpora l'opzione LFORMAT. Ammettiamo che tu voglia cancellare in una directory tutti i file che sono stati creati dopo una determinata data.
Normalmente dovresti digitare:

```
List Nomedir SINCE gg-mmm-aa
```

dopodiché dovresti armarti di pazienza e di buona volontà e cancellare tutti i file uno alla volta.
Con l'opzione LFORMAT questa operazione diventa molto semplice, digitando:

```
List >RAM:Pippo Nomedir SINCE gg-mmm-aa LFORMAT="Delete %s"
```

otterrai un file di nome Pippo che conterrà:

```
Delete file1
Delete file2
Delete file3
.
.
```

In pratica LFORMAT stampa sul video (nel nostro caso in un file, visto che abbiamo rediretto l'output) la stringa richiesta, sostituendo ad "%s" il nome del file. A questo punto basterà eseguire il file con "Execute pippo" per cancellare tutti i file desiderati in un sol colpo.

Ancora Virus !

Spett. Amiga Magazine,
sono un possessore di Amiga 500 che recentemente ha avuto problemi con vari tipi di virus. Ultimamente mi è capitato un virus che produce un messaggio in inglese del tipo: "un virus è una malattia,..., la pirateria è un crimine e questa è la cura: BGS9....". Non sono riuscito a toglierlo nemmeno con i migliori antivirus. Cosa posso fare?

Filippo De Tomasi

I tuoi dischetti sono infetti dal virus BGS9, che è un virus del tutto simile all'IRQ. Questo tipo di virus è uno dei pochi che non si installa nel bootblock, ma si sostituisce al primo comando presente nella startup sequence e si inserisce nel primo hunk. Per eliminare questo virus ti consigliamo di utilizzare il programma KV, Kill Virus che è un antivirus distribuito con VirusX.

Dalla redazione...

Per ovvie ragioni di spazio non ci è possibile pubblicare tutte le richieste che giungono in redazione.
Cercheremo di rispondere privatamente alle lettere non pubblicate.

Collaboratori ?

La redazione invita tutti i lettori ad inviarci i propri programmi e si riserva di pubblicare e retribuire quelli che ritiene i migliori. Sono ben accetti sia programmi di giochi, sia programmi di utility.
I programmi inviati non saranno, per ragioni redazionali, restituiti.

Le opere dovranno essere inviate a:
Gruppo Editoriale Jackson
Amiga Magazine
Via Pola, 9
20124 Milano

International User Group

User Group del mondo unitevi !! Questo è il motto che ha la nuova rubrica di Amiga Magazine. Se fate parte di un User Group e volete vedere pubblicato il vostro nome, inviateci i vostri dati: nome e il nome del vostro gruppo, indirizzo, telefono Bbs ecc. Vengono pubblicati anche indirizzi di User Group stranieri in modo da aprire le frontiere fra utenti Amiga. La redazione non si assume nessuna responsabilità sui dati inviati e pubblicati.

Il nostro indirizzo è:
**Gruppo Editoriale Jackson
Amiga Magazine - Area Consumer
Rubrica "International User Group"
Via Pola, 9 - 20124 Milano**

U.S.A.

Valley Video Workshop, 2013 Los Feliz #3, Thousand Oaks, CA 91362

San Fernando Valley Amiga Users Group (SFAVAUG), P.O. Box 8183, Van Nuys, CA 91406

South Bay Commodore Users Group. P.O. Box 1899, Chula Vista, CA 92012-1899

San Diego Amiga Users Group, P.O. Box 80186, san diego, CA 92138

Club-64 (San Bernardino), P.O. Box 514, Patton, CA 92369

Amiga Friends, P.O. Box 4186, Huntington Beach, CA 92605

South Orange Commodore Klub (SOCK), 25401 Champlain Rd., laguna Hills, CA 92653

Ventura Amiga Users Exchange, P.O. Box 2738, Ventura, CA 93010

A Bakersfield Area Commodore Users Society (ABACUS), P.O. Box 40334, Bakersfiled, CA 93306

Simply Users of Computers Combining Experience for strenght and Success, 184 Santa Ynez Ave.,Paso Robles, CA 93446

Flashfire Mouse & Joystick Sensor

Reggetevi forte! Ecco una novità veramente sensazionale per i numerosi videoplayer italiani, ma anche per tutti coloro che usano il computer per applicazioni grafiche. In Italia ancora poco conosciuto, all'estero già diventato per molti un accessorio indispensabile.

a cura della redazione

La vita degli appassionati e dei professionisti in computer art game diviene ogni giorno più facile grazie alle rivoluzionarie innovazioni tecniche. *Il Mouse & Joystick Sensor* è frutto di sofisticate tecnologie. E' questo un accessorio definito da molti indistruttibile, in quanto non contiene parti meccaniche in movimento soggette a rotture in seguito ad usura nel tempo, infatti, non solo è privo di leveraggi come cloche e perni di pressione metallici o plastici, ma non prevede neppure l'impiego dei microswitch e dei relativi tasti mobili. Le comodità che offre questo accessorio sono intuitive soprattutto per chi ama giocare col computer, in quanto è possibile comandare le funzioni usando una sola mano. Giocando a Barbarian oppure a Kick Off, potrete tranquillamente bervi il solito tè!

Mouse & Joystick Sensor è interamente elettronico in quanto impiega circuiti integrati digitali CMOS con tasti sensitivi. Funziona come joystick e come mouse su 360°. Per farlo entrare in funzione è sufficiente sfiorare i comandi, i quali reagiscono al semplice contatto con le dita umane; gli stessi non devono essere mai premuti perché l'apparecchio è sensibile al contatto della pelle. A differenza dei suoi simili attualmente in commercio, ha una sua originalità particolare, funziona ad induzione capacitiva attraverso il corpo umano.

I quattro contatti a forma di quadrifoglio sono i punti direzionali. Gli altri due contatti, nel caso del game, devono essere utilizzati uno per il fuoco lento e l'altro per il fuoco rapido. Chiaramente tutte le sue funzioni vengono realizzate mediante un sensor. *Flashfire Sensor* è dotato della classica presa joystick universale DB9, compatibile con le relative spine montate sulla maggior parte degli home computer.

Siamo abituati a vedere il joystick sotto una veste come un insieme di pulsanti. *Flashfire* a sensori, invece, è dotato nel proprio interno di circuiti elettronici in grado di trasformare gli impulsi elettrici, forniti dal contatto dei sensori, nei segnali normalmente presenti sui terminali della presa joystick. Elettronicamente parlando, la resistenza introdotta dal corpo umano tra la placchetta che fa da sensore e la massa, formata dal rivestimento metallico che ricopre il resto del contenitore, provoca, attraverso l'azione di circuiti di commutazione inseriti nei chip, la chiusura a massa del relativo terminale

come se questo fosse stato commutato da un qualsiasi contatto meccanico. Il consumo del circuito, trattandosi di componenti CMOS dall'assorbimento bassissimo, è irrilevante e l'energia necessaria all'alimentazione viene "attinta" direttamente dal computer attraverso la stessa porta joystick.
Gli appassionati di videogame che hanno avuto l'occasione di provare *Flashfire*, hanno dichiarato che le sensazioni d'uso rispetto ai joystick tradizionali sono differenti, in quanto si ha una migliore prontezza, una più tempestiva reazione, maggior comodità e precisione, specialmente nei giochi dove queste doti sono essenziali.
Naturalmente, come per tutte le innovazioni, è necessario fare un minimo di pratica per sfruttare efficacemente questo bellissimo accessorio. L'assenza di organi meccanici elimina ogni tipo di ritardo dovuto allo spostamento dei contatti e la sensazione tattile è molto gradevole, in quanto ci si accorge di avere tra le mani un controllo molto sensibile sullo strumento. Non dimentichiamo, inoltre, che l'assenza di leve e contatti rende *Flashfire Sensor* pressoché indistruttibile, a meno che non venga trattato in modo assolutamente inusuale (col martello tanto per fare un esempio).
Usando assiduamente questo futuristico joystick, inoltre, avrete modo di scoprire che è adattissimo e utilissimo come accessorio per le applicazioni grafiche. La pratica vi porterà ad entrare in simbiosi con il quadro a sensori e disegnare a video diventerà di una tale semplicità, che sarà come farlo sulla carta. Per i videogame gioca nettamente a favore la velocità di risposta del sistema a sensori. Lo potrete notare specialmente nei giochi sportivi a squadre, per esempio calcio, rugby, basket e nei giochi di guerra come Rambo, Operation Wolf e tantissimi altri...
Flashfire a sensori, dunque, è una novità assoluta che contribuisce a migliorare le prestazioni del vostro computer anche come mouse. Non avendo la possibilità di spostarsi su di un tavolo come il classico mouse, non necessita di una grande superficie per il suo utilizzo come strumento di lavoro. ▲

Flashfire Mouse & Joystick Sensor è prodotto dalla:

GP Elettronica
Via IV Novembre, 32/34
20092 Cinisello Balsamo (MI)
Tel. 02/6189551
Fax. 02/66012023

ed è disponibile presso

Electronics Performance
Via S. Fruttuoso, 16/A
20052 Monza (MI)
Tel. 039/744164

AMIGA MAGAZINE

Next...

...sul prossimo numero

▲ **Le favolose pagine di Transactor**
Ovvero le pagine del programmatore

▲ **Speciale Grafica**
Prova pratica di strumenti grafici e di programmi per disegnare

▲ **Nuovi prodotti**
Il nuovo Amiga 3000 e il WorkBench V.2.0

▲ **Calligrapher**
Un disegnatore professionale di caratteri

▲ **3-Demon**
Un programma per la creazione di oggetti

▲ **Programmazione facile**
Costruiamoci un videogioco

Appuntamento a luglio in edicola !!

Il Workbench

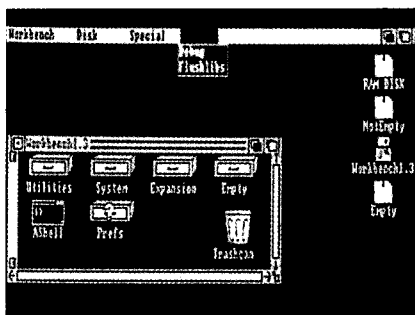
John Foust

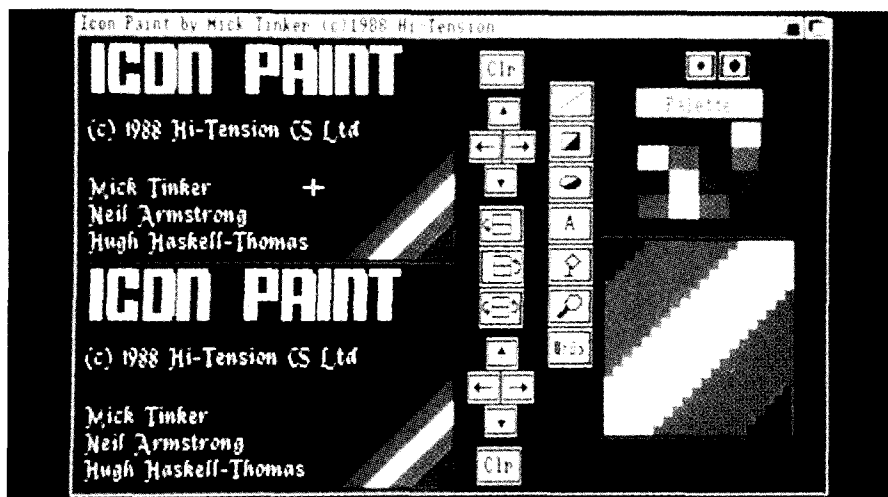
Se avete un Amiga, avete il Workbench, e "Spotlight" vi aiuta a ottenere di più dal software e dall'hardware che già possedete per il vostro Amiga. Questo mese, discuterò di alcuni problemi comuni che avreste potuto avere con il Workbench, oltre a darvi alcuni consigli e ad esaminare programmi di public domain, al fine di aiutarvi ad usarlo con maggior efficacia.
Il sistema operativo di Amiga ha due facce: il Workbench e il CLI. Il Workbench si controlla mediante il mouse, la selezione delle icone e i gadget. Teoricamente, il Workbench esegue tutto quello che serve a controllare ogni aspetto di Amiga, come far partire un programma, adattarlo alle proprie esigenze, gestire file e dischi. Puntare e selezionare è più facile che usare il CLI. Sfortunatamente, non tutti i programmi lavorano bene con il Workbench e alcune cose non funzionano con il CLI. Con il CLI, tutto viene inserito mediante tastiera in brevi stringhe chiamate linee di comando (CLI sta per Command Line Interface, interfaccia per la linea di comando). Per usare il CLI in maniera adeguata, dovete impegnarvi per memorizzare la sintassi di ogni comando. Il CLI fornisce un controllo più sottile del sistema, una volta che abbiate imparato ad usarlo.
Una buona guida al Workbench è l'Introduzione ad Amiga della Commodore, che accompagna ogni macchina. Vi si discutono i concetti fondamentali del puntare-e-selezionare e i dettagli su come usare programmi quali il Notepad.
Sia il CLI che il Workbench fanno affidamento sui file. I file contengono insiemi di dati; testi, programmi e fogli elettronici sono tutti immagazzinati in file. Perché un'icona appaia sul Workbench, ci devono essere su disco due file. Se abbiamo un file di dati di un foglio elettronico chiamato "Tasse", allora deve esistere sul disco un secondo file chiamato "Tasse.info". Questo secondo file, detto file info, contiene l'immagine dell'icona che viene visualizzata, assieme ad informazioni sulla posizione che questa deve assumere sullo schermo. Quando, per esempio, trasportate l'icona "Tasse" in un drawer (cassetto) chiamato "1989", il Workbench muove entrambi i file nel drawer. Anche i drawer fanno affidamento sui file info. Se un drawer non possiede un file info che gli corrisponda, non potete manipolarlo attraverso il Workbench, perché questo non riesce a vederlo.
Non avere un'icona per ciascun file porta a un problema comune. Dischi che appaiono vuoti possono restituire l'errore "Disk Full" (disco pieno), quando tentate di salvarvi qualcosa. Se fate affidamento sulle icone per stabilire la presenza di file sul disco, rimarrete ingannati. Se esaminate il disco attraverso un file requester (come quello in Deluxe Paint), riuscirete a trovare i file che occupano lo spazio. La maggior parte dei file requester mostrano ogni file presente sul disco. Su un disco Workbench nuovo, esistono molti file e drawer che non hanno icone.
Potete rimuovere questi file in due modi. In primo luogo, imparare ad usare alcuni comandi CLI, come cd, dir e delete. Non avete bisogno di conoscere altri comandi CLI per rimuovere file inutili da un disco.

Ogni file o drawer può essere manipolato attraverso il CLI. Il CLI non dipende dai file info. In secondo luogo, usare un programma come CLImate, che presenta un file requester con pulsanti che vi consentono di cancellare file (CLImate permette anche di copiare e muovere file, oltre a svolgere altri compiti collegati al disco). Esistono dozzine di utility di public domain che svolgono lo stesso compito, tutti noti col nome generico di DirUtil come ClickDos e Browser [DirUtil è anche il nome specifico di un programma di PD, N.D.T.]. C'è anche un altro comune errore di comprensione riguardo ai dischi pieni. Trascinare un'icona nel Trashcan (bidone della spazzatura) non significa eliminare il file dal disco, ma spostarlo nel drawer del Trashcan. Lo spazio che occupa sul disco non viene liberato per altri usi, finché l'icona non viene effettivamente cancellata. Ciò vi permette di recuperare un file dal Trashcan, se l'avete gettato via accidentalmente. Sullo schermo del Workbench ci sono più menu. Il secondo menu è chiamato "Disk" e ha un menu item chiamato "Empty Trash" (buttare la spazzatura). Questa opzione cancella tutti i file nel drawer Trashcan. Per cancellare un

IconEd è un programma che trovate nel disco Extras. Con esso è possibile fare o modificare le icone dei vostri file

file direttamente, al fine di liberarne lo spazio, si selezioni l'icona (con una singola pressione sul pulsante del mouse) e si selezioni l'opzione "Discard" (scartare) del menu "Project". Immediatamente, la lunghezza del file sarà aggiunta allo spazio libero sul disco. Muovendo file e drawer, potreste notare un bug poco noto del Workbench. Qualche volta, quando state trascinando un'icona o un gruppo di icone, queste scompaiono quando rilasciate il pulsante del mouse. Non sono realmente scomparse; sono state spostate nel drawer, alla sinistra della posizione in cui le avete rilasciate. Se aprite quel drawer, troverete le icone mancanti. Nessun danno è stato fatto. Riportatele nel luogo in cui volevate metterle e fissatele in quella posizione con la voce "Snapshot" (fare un'istantanea) del menu "Special" del Workbench.
Se tutto quello di cui un file ha bisogno, per diventare visibile al Workbench, è un'icona, come potete realizzare icone per i file? Il disco Extras include un programma chiamato IconEd che può essere usato per creare e modificare icone. Ci sono alcune complicazioni. Le icone sono di diversi tipi. Per fare una nuova icona, dovreste duplicarla a partire da un'altra icona dello stesso tipo. I programmi necessitano di icone "Tool" (strumento), mentre i file di dati usano icone "Project" (progetto). Le icone dei dischi, dei drawer e

del Trashcan sono, a loro volta, dei tipi distinti. Per realizzare un'icona per un file, si carichi un'icona del tipo giusto, si disegni l'immagine e poi la si salvi usando il nome di quel file. Si noti che le icone dei dischi devono essere chiamate "Disk" con una D maiuscola; la d minuscola non funziona.
IconEd aggiunge automaticamente il suffisso .info per voi. IconEd non ha una gamma molto ampia di funzioni per disegnare. Con gli strumenti adatti, potrete realizzare icone mediante qualsiasi programma per disegnare. Si tenga presente che quello del Workbench è uno schermo che puo' arrivare alla risoluzione di 640 x 400 pixel [640 x 512 PAL, N.D.T.] con quattro colori. I programmi di public domain vi verranno in soccorso. Il disco 85 di Fred Fish contiene un buon insieme di strumenti adatti allo scopo. Uno converte un pennello (brush) in un'icona, un altro converte un'icona in un pennello e un altro ancora riduce ogni immagine delle massime dimensioni IFF in un'immagine più piccola a quattro colori. Il disco 55 di Fred Fish contiene un programma per catturare schermi che è in grado di salvare ogni schermo Workbench come file IFF; così potrete importarlo in un programma per disegnare al fine di ricavarne i colori che usate normalmente sul Workbench. Si noti che avete bisogno di usare il CLI per navigare in un Fish Disk e lanciare questi programmi. Fare un'icona per un file non garantisce che esso possa lavorare correttamente col Workbench. Molti programmi sono basati sul CLI e dipendono da input e output testuali. Esistono programmi di public domain che permettono di lanciare programmi CLI dal Workbench, ma non è detto che funzionino con tutti i programmi. La voce "Info" (informazioni) del menu "Project" del Workbench fornisce informazioni vitali su un'icona e il suo file corrispondente. Selezionate un file, scegliete "Info" e apparirà una finestra. In alto a sinistra, vedete il tipo di icona, come "Project" o "Disk" e la lunghezza del file. Subito sotto, il gadget "Comment" (commento) vi permette di introdurre un breve commento sul file. Il gadget "Tool Types" (simboli per lo strumento) mostra informazioni aggiuntive che vengono passate al programma quando viene eseguito.
Se il tipo di icona è "Project" o "Disk", compare anche un campo "Default Tool" (strumento di default). Quest'area contiene il nome completo

del programma che ha creato questo file. Quando si seleziona due volte un'icona "Project", viene lanciato il programma il cui nome è contenuto in "Default Tool". Per esempio, l'icona per un'immagine fatta con Deluxe Paint" ha in "Default Tool" la stringa "DPaint:Dpaint". La vostra doppia selezione fa iniziare la ricerca del file "DPaint" su un disco dello stesso nome.
La duplicazione dei file genera un problema collegato al gadget "Default Tool". Quando un'icona viene duplicata, la nuova copia avrà il nome del vecchio file con l'aggiunta delle parole copy of (copia di). Se voi lasciate questa copia del programma sul disco e cancellate il programma originale, improvvisamente le icone "Project" smettono di funzionare. Quando fate una doppia selezione, non riescono più a trovare il programma che le ha generate. Se avete lasciato un file chiamato copy of Dpaint sul disco, le icone "Project" per i disegni fatti con Deluxe Paint hanno un "Default Tool" che contiene "DPaint:Dpaint", invece di "DPaint: copy of Dpaint". Dopo avere usato "Duplicate" (duplicato), assicuratevi di rimuovere il prefisso copy of dal nome del file, usando la voce "Rename" (rinominare) del menu "Project" del Workbench.
Si noti che questo accade anche ai nomi di dischi, se vengono copiati usando la funzione "Duplicate" del Workbench. Un'applicazione perfettamente integrata al Workbench non dovrebbe avere questi problemi, ma la maggior parte dei programmi non sono perfetti. Se preferite usare esclusivamente il Workbench e non siete soddisfatti di programmi che non funzionano correttamente con il Workbench, scrivete una lettera al produttore del software. Le compagnie che lavorano per Amiga sono ancora piccole e abbastanza sagge da dare ascolto al feedback ben meditato che proviene dagli utenti.
Potreste tentare questo trucco per aggiungere un menu ulteriore allo screen del Workbench. Da CLI, inserite la linea LOADWB -debug (l'opzione debug deve essere digitata in minuscolo) e premete Return. Il Workbench ridisegnerà se stesso e un nuovo menu apparirà sotto il testo dell'indicatore di memoria. Il menu ha due voci: "debug" e "Flushlibs". Non selezionate "debug" se non avete un secondo Amiga connesso alla porta seriale. Amiga si bloccherà, il mouse non si muoverà e non ci sarà altra scelta che il reset. Questa voce lancia un programma built-in di debug per programmatori, che opera attraverso la porta seriale a 9600 baud. Per usarlo avete bisogno di un computer o un terminale sulla porta seriale.
La seconda voce, "Flushlibs" è più utile. Se tenete sott'occhio l'indicatore della memoria sul Workbench, prima e dopo aver lanciato un programma, noterete che non tutta la memoria usata è stata liberata. Quando un programma lavora, può caricare font o librerie di funzioni in memoria. Queste possono rimanere in memoria dopo che il programma sia terminato. L'opzione "Flushlibs" vi permette di liberare questa memoria. Con grosse font, ciò può significare una consistente quantità di memoria. Per risultati migliori, selezionate questa voce più volte consecutivamente, per aggirare un piccolo bug del sistema operativo. Ancora, programmi imperfetti non restituiranno tutta la memoria. Lanciare più volte questo tipo di programmi potrebbe condurre a una visita del Guru.
L'estate scorsa, la Commodore ha rilasciato l'Enhancer 1.3, una nuova versione del sistema operativo. Il vostro fornitore di prodotti per Amiga lo vende per circa 90.000 lire. Comprende un breve manuale che spiega le innumerevoli opzioni di tutti i comandi CLI, incluse quelle nuove. Francamente, anche come piccolo Guru di Amiga, ho trovato la documentazione deprimente. La maggior parte del manuale è povera di dettagli sui comandi CLI.
Il Workbench non si trova neppure nell'indice, ma la versione 1.3 porta almeno una novità che dovrebbe affascinare gli utenti del Workbench. Un programma chiamato IconMerge aiuta a creare icone con immagine doppia. Sono le icone la cui immagine muta quando le selezionate, come l'icona del Trashcan, in cui il coperchio si apre quando la si seleziona. IconMerge vi permette di separare un'icona doppia già esistente in due immagini e di fondere due icone in un'icona doppia.
L'Enhancer 1.3 comprende altre nuove funzioni, come un RAM disk che sopravvive al reset, metodi più facili per spostare printer driver e opzioni di Preferences in nuovi dischi sotto controllo del Workbench e, soprattutto, un più veloce sistema di accesso ai dischi per gli utilizzatori di hard disk. ▲

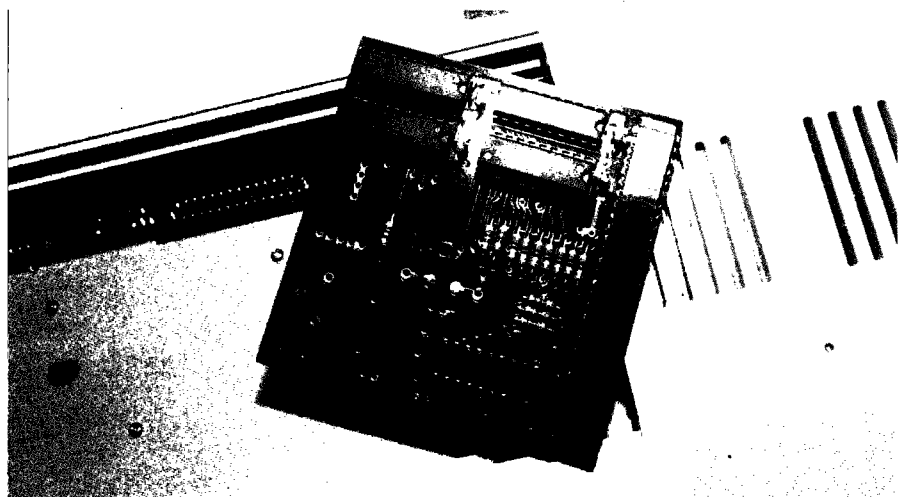
La schermata principale di Icon Paint, il primo programma a permettere icone con 16 colori

Espansioni di memoria su Amiga 500

Analizziamo alcuni dei molteplici problemi generati dall'installazione di questi semplici Add-on

a cura di Computer Lab
Via Cadore, 6 - Milano

Chi possiede un Amiga 500 potrà certamente confermare che il primo optional acquistato, o da acquistare, è l'espansione RAM per portare la dimensione della memoria ad 1 Mb. Inutile soffermarci all'elencazione delle nuove possibilità di lavoro offerte da questa semplice board dal costo oscillante tra le 200.000 lire per i prodotti non originali e le circa 300.000 lire per il prodotto originale Commodore siglato A-501 dotato di Real Time Clock e garanzia di un anno.

Il mercato

Esistono, come avviene per quasi tutti gli Add-on Amiga, diversi produttori che soddisfano la richiesta del mercato, i quali, per la produzione delle board, utilizzano componentistica e soluzioni tecnologiche differenti fra loro. Ciò che ci conforta è rappresentato dal fatto che, per quanto riguarda queste espansioni, la quasi totalità dei prodotti non originali è "Made in Italy", dato questo che conferma la dinamicità delle nostre piccole aziende artigianali operanti nel settore elettronico. Visionando i prodotti a catalogo nei negozi più riforniti possiamo circoscrivere ad un massimo di quattro i modelli più significativi: non ritenendo corretto citare i produttori ci limitiamo a elencarne le differenze essenziali. Per due modelli ci sentiamo in dovere di dare un giudizio positivo sia per la "pulizia" dello stampato, sia per il fatto che l'espansione è alloggiata in un apposito contenitore molto utile nella fase di installazione per i meno esperti (analizzeremo più avanti questo aspetto) e indispensabile per garantire un corretto alloggiamento anche in fase di spostamento di Amiga. Entrambe le espansioni sono "costruite" facendo uso di RAM dinamiche da 4 x 256 Kb limitando così il numero dei chip RAM a 4 per formare un banco da 512 Kb. Esiste poi un terzo modello, analogo ai precedenti ma di costruzione palesemente "artigianale", senza contenitore e senza il circuito

Un modello di espansione non originale

L'espansione A-501 senza il contenitore metallico

dell'orologio. L'ultimo modello analizzato non fa uso delle già citate RAM da 4 x 256 Kb ma delle quasi obsolete 1 x 256 Kb in quantità di 16 montate su IC-socket: il maggior numero di RAM e relativi chip di indirizzamento determinano le dimensioni della espansione che risulta raddoppiata rispetto ai precedenti modelli citati.
A-501 è la sigla assegnata da Commodore alla propria espansione RAM che si presenta racchiusa nel contenitore metallico con l'ormai famoso sigillo garanzia ben in vista. Le dimensioni non sono certamente ridotte anche se i modelli in distribuzione da circa un anno montano anch'essi RAM da 4 x 256 Kbyte: le funzioni del Real Time Clock sono garantite dal classico integrato 6242 supportato da un quarzo e relativa batteria al Nichel Cadmio.

La connessione

Per chi con l'elettronica ha ben poco in comune potrebbe risultare inutile approfondire l'argomento leggendo il seguito di questo articolo; non siamo ovviamente d'accordo con questo pensiero perché riteniamo che la conoscenza seppur minima del principio di funzionamento del computer che usiamo tutti i giorni serva per porsi nella condizione di salvaguardarci da errate operazioni e in caso di guasto ci permette di meglio comprendere l'eventuale natura del guasto stesso o, se fosse necessario, replicare a false dichiarazioni di falsi riparatori.
A questo punto, per meglio seguirci, prendete il vostro manuale "Introduzione" consegnatovi all'atto dell'acquisto di Amiga (e forse già archiviato) e portatevi alla pagina F-5 dell'appendice.
Quasi sicuramente uno schema elettrico risulterà meno interessante di una qualsiasi altra pagina riportante la sintassi di un comando Amiga Dos, ma è necessario leggerlo.
Al centro della pagina, in alto, individuate la sigla CNX assegnata al connettore che permette la connessione delle espansioni di memoria su Amiga 500: sul disegno il connettore è rappresentato con le due file di contatti pari e dispari, numerati da 1 a 56. Prendete ora il vostro Amiga 500, scollegatelo e dopo averlo rovesciato aprite lo sportellino dell'alloggiamento dell'espansione: vedrete il connettore CNX e, rispetto allo schema elettrico, dovete considerare che la numerazione parte dal basso, mentre la fila dei contatti pari è la più interna. Teniamo ben presente questa disposizione fisica, ci servirà più avanti.
Ritornando allo schema elettrico iniziamo l'analisi con l'individuazione delle linee di alimentazione: il ground GND, meglio conosciuto come massa è riportato sui pin 3, 4, 21, 22, 33, 34, 53, 54; i livelli alti delle tensioni sono: i + 5 volt (nello schema indicati come Vcc quale alimentazione degli integrati) li troviamo ai pin 1, 2, 51, 52. Abbiamo poi disponibili altre due tensioni continue, non sempre usate (ma sempre presenti sul connettore): + 12 volt sul pin 55 e - 12 volt al pin 56. Sui pin dal 5 al 20 troviamo la linea del Data Bus, linea a 16 bit di dialogo diretto con il processore 68000 mentre ai pin numerati dal 23 al 31 è indirizzata la linea a 9 bit del Multiplexed Addresses (indirizzamento memoria). Abbiamo poi presenti alcuni segnali di clock e interrupt.

Le connessione errate

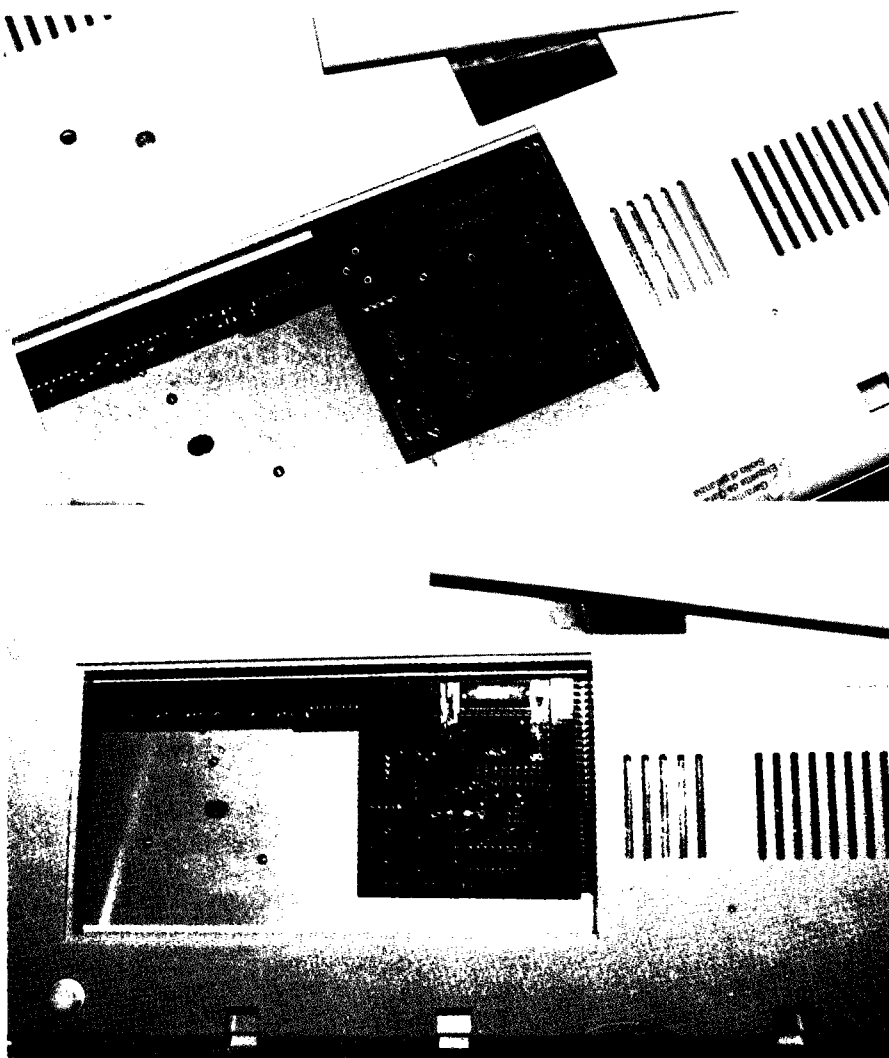
Visto l'insieme dei segnali e tensioni che troviamo disponibili sul connettore CNX ci risulta ora più semplice simulare (in teoria) gli effetti causati da una connessione• errata della espansione RAM. Innanzitutto prendiamo in considerazione il caso, abbastanza inusuale, di connettere una espansione di memoria guasta: fin tanto che il guasto così definito non comporta dei corti circuiti tra le diverse linee il risultato è di non ottenere la disponibilità del Mega RAM ma solo della RAM di base.
Ma cosa capita se, ad esempio, in fase di saldatura dei componenti è stato causato un corto circuito tra una pista dei 12 volt con la linea dati o con i 5 volt? La nostra board di espansione si trasforma, nostro malgrado, in una sorta di ponte che reinvia una tensione proveniente da Amiga sul bus di Amiga stesso. Il danno si ha ovviamente quando la tensione reindirizzata è di 12 volt poiché si tenga presente che i dati vengono distinti in 0 bassi e 1 alti corrispondenti rispettivamente ai livelli di tensione 0 e 5 volt.
Questa causa di guasto pur essendo originata dall'aver connesso l'espansione non è comunque imputabile al cliente che paradossalmente si trova ora con due prodotti guasti: Amiga 500 e ovviamente l'espansio-

ne "killer". Nel caso in cui l'espansione è un prodotto originale Commodore, nuovo, con garanzia originale Commodore Italiana, nessun problema: o meglio i problemi li incontreremo noi di Computer Lab o altro centro assistenza autorizzato a cui vi sarete rivolti ma nel caso in cui l'espansione non è la classica A-501 il problema deve essere riportato al punto vendita che, in funzione della propria professionalità, risponderà risolvendovelo o malauguratamente addossandovi le responsabilità dell'accaduto.
Diverso è il caso in cui il guasto si sia verificato perché l'espansione è stata erroneamente inserita o inserita a computer acceso. Quest'ultima situazione seppur impensabile da molti è realmente accaduta e per dovere di cronaca vi confermiamo che a farne le spese è stato sia Amiga 500 che l'espansione. Più ricorrente, invece, il caso di inserimento errato: le espansioni non originali, senza contenitore, possono essere inserite capovolte (con i componenti rivolti verso l'esterno) o spostate verso l'alto (sempre osservando Amiga capovolto) di un certo numero di pin. Vediamo a livello circuitale cosa capita: nel primo caso connettiamo la linea dei + 5 volt su entrambi i 12 volt, negativi e positivi ma, in quanto come già visto i 5 volt sono presenti sui pin 1 e 2 mentre i + e - 12 volt ai corrispondenti 55 e 56. Così facendo alimentiamo l'espansione a 12 volt! Altro aspetto meno distruttivo è rappresentato dal fatto che creiamo una sorta di miscelazione dei segnali del Data Bus proveniente dal 68000 con il Multiplexed. Nel secondo caso, connettendo l'espansione spostata, ad esempio, di un pin verso l'alto otteniamo una connessione dei 12 volt sul GND mentre la linea dei 5 volt coincide con il GND di Amiga e i 12 volt presenti sul GND vengono reinviati sulla linea dati di Amiga. Il risultato finale di entrambe le errate connessioni è una variabile più o meno alta di integrati bruciati, in relazione al tempo di accensione di Amiga 500 servito per convincerci che qualche cosa non funzionava. Se vi resta del tempo e voglia provate a simulare, sempre teoricamente e analizzando lo schema elettrico, le diverse possibilità e fateci sapere cosa ne pensate.

Necessarie conclusioni

Molto spesso si è portati a pensare che connettendo semplici Add-on ai nostri computer anche in caso di errata connessione non si possono verificare gravi danni: purtroppo questa è una affermazione errata o quantomeno semplicistica che possiamo tranquillamente smentire con i dati che estrapoliamo giornalmente dalle nostre schede di lavoro relative alle riparazioni effettuate per ripristinare danni causati da "semplici installazioni". Sono sicuramente semplici se prima di accendere e mettere in funzione le nostre apparecchiature usiamo a quest'ultime la cortesia di effettuare almeno due controlli: perderemo qualche minuto in più ma sicuramente eviteremo noiosi e dispendiosi viaggi verso centri di assistenza.
Nella scelta di quest'ultimi, non ci stanchiamo di ripeterlo, preferite le strutture autorizzate da Commodore Italiana e non dal rivenditore che sempre più spesso si improvvisa riparatore. ▲

Connessione corretta di una espansione non originale

Classica connessione errata

Novità dalla Newtronic: Videon 3.0 e ...

Daniele Cassanelli

Nei primi giorni di Aprile la Newtronic Technologies ha presentato a Rapallo, presso l'Hotel Lucciola, le sue ultime novità hardware & software nel campo video.
Il fiore all'occhiello è la nuova versione del digitalizzatore Videon 3.0; seguono un nuovo Genlock completamente riprogettato, Gen2, una versione del Videon per PC IBM equipaggiati con una VGA (di cui noi non parleremo, naturalmente!), e per finire, Videobox, un apparecchio dalla caratteristiche esclusive.

Videon 3.0: 27192 colori reali !

Purtroppo la versione hardware non era ancora disponibile (lo sarà verso Novembre-Dicembre) per cui alla presentazione abbiamo potuto solo ammirare l'aggiornamento software, che, diversamente da come ci aspettavamo, "gira" anche sul vecchio hardware.
Le novità riguardano, soprattutto, i nuovi metodi di acquisizione dell'immagine: il più interessante viene denominato "fast scan" attraverso il quale è possibile digitalizzare un'immagine lo-res in soli 13 secondi. Grazie al nuovo hardware, munito di nuovi ADC (Analog Digital Converter) a 8 bit per canale, sarà possibile digitalizzare immagini sfruttando una palette teorica di 16.7 milioni di colori.
Un altro metodo di acquisizione è detto "slow scan" o di acquisizione multipla, con il quale è possibile ottenere immagini nitidissime anche quando si usano sorgenti video di scarsa qualità, come, ad esempio, il segnale prodotto dal fermoimmagine della stragrande maggioranza dei videoregistratori.
Altre novità nei metodi di acquisizione delle immagini sono: "Antialiasing", che, grazie ad uno speciale algoritmo di filtraggio, elimina le scalettature e i difetti delle immagini e "Manual" che permette di acquisire immagini digitalizzate direttamente da telecamere con uscita RGB, oppure di digitalizzare a colori anche da telecamere in bianco e nero (utilizzando l'ormai nota tecnica dei filtrini colorati).
Inoltre, grazie al tool "Fix Palette", è possibile digitalizzare immagini sempre con la stessa palette di colori, particolare utile quando si devono usare le immagini digitalizzate per animazioni.
Restando sempre in tema di animazioni, il nuovo software di Videon 3.0 permette di creare animazioni "mappando" fino a quattro immagini diverse su uno stesso solido. A questo punto l'animazione creata potrà essere salvata in formato Anim, formato supportato da Videoscape, ANIMagic e altri programmi quali Sculpt-Animate 4D e Turbo Silver, con il quale è stato realizzato un'animazione demo stupenda. Tra i nuovi metodi di salvataggio spicca la possibilità di salvare l'immagine digitalizzata in RGB in formato IFF a 24 bit (16.7 Milioni di colori!), naturalmente quando sarà disponibile la nuova versione del digitalizzatore.
Passiamo ora ai tre nuovi metodi grafici supportati: il primo, denominato Super-Hires, corrisponde al "Dinamic Mode" del Digiview 4.0: è un modo pseudo-HAM che permette di digitalizzare e visualizzare immagini a 4096 colori in alta risoluzione. Ma la vera novità è il "27000 Mode" che permette di visualizzare 27192 colori reali sul nostro monitor, e, grazie a particolari tecniche di dithering, si potranno visualizzare fino a due milioni di colori apparenti. Questo metodo di visualizzazione sarà utilizzabile sempre con le nuova versione del digitalizzatore quando digitalizzeremo a 24 bit (8 bit per componente di colore).
La tecnica utilizzata per visualizzare tutti questi colori è, se vogliamo, lo stesso metodo utilizzato per visualizzare uno schermo interlacciato. Così come vengono visualizzate prima le linee pari e poi quelle dispari in uno schermo interlacciato, nel "27000 mode" vengono alternati diversi screen ad una velocità tale che l'occhio umano non percepisce il cambiamento..
L'ultima novità tra i nuovi metodi grafici supportati dal nuovo software riguarda la possibilità di digitalizzare in modo EHB, Extra Half Brite, a 64 colori. Su questo metodo di visualizzazione, oltre che al "27000 mode", che la Newtronic ha focalizzato i nuovi sforzi, ottenendo risultati seriamente paragonabili all'HAM.
Abbiamo accennato più volte al nuovo digitalizzatore; vediamo quali sono le novità: innanzitutto i nuovi ADC a 8 bit che permette di digitalizzare immagini a 24 bit. Sarà equipaggiato di serie con un ingresso Super-VHS che permetterà di acquisire immagini direttamente da apparecchiature con questo tipo di uscita.
L'ultima novità riguarda l'aggiunta di un traking regolabile, ossia di poter sincronizzare manualmente il digitalizzatore con la sorgente video con lo scopo di eliminare tutti i disturbi che possono compromettere la qualità dell'immagine.

Gen2: un nuovo genlock...

Completamente riprogettato e riveduto ha una peculiarità esclusiva nel suo settore: una volta collegato ad Amiga di sincronizza automaticamente al computer, tallone di achille di tutti i genlock. Anche questo prodotto era, per usare un termine utilizzato per il software, in beta release.

Videobox: il top

Videobox è un apparecchio rivoluzionario nel suo genere: è allo stesso tempo genlock, digitalizzatore e frame-grabber in tempo reale. L'immagime viene digitalizzata con ADC a 6 bit per un totale di 262144 colori (esiste però la possibilità di espandere ulteriormente la macchina arrivando a 8 bit per canale raggiungendo i 16 milioni di colori) e viene memorizzata in tempo reale nel buffer interno. Avrà la possibilità di utilizzare ben tre genlock contemporaneamente.
Lo scopo della Newtronic non è comunque quello di creare un apparecchio per Amiga, ma è quello di creare un nuovo dispositivo collegabile a qualsiasi computer (naturalmente con il software che lo supporti).
Alla fine della presentazione la Newtronic ha promesso di mandarci il nuovo software e, appena possibile la nuova versione del digitalizzatore, per cui spero al più presto di avere il piacere di vedere 27000 colori sul mio monitor e, naturalmente, di potervene parlare. ▲

Per ulteriori informazioni:
Newtronic
Via Cantore, 50 Genova
Tel.010/416570

Articolo tradotto dai n.4, 5
Ottobre-Dicembre 1989
di Compute!'s Amiga Resource

Tutto quanto fa BASIC

Il BASIC è il più popolare linguaggio per computer mai realizzato e da diversi anni sta cercando di sfuggire alla sua immagine di strumento per principianti. Ci riuscirà su Amiga?

Dale McBane

Come la maggior parte dei linguaggi, il BASIC è nato per necessità. All'inizio dell'era informatica era veramente difficile, per carenza di macchine disponibili, usare il computer. Costava più usare i computer che scrivere il software. I programmi venivano preparati separatamente, per esempio perforando apposite schede e mandati in elaborazione come lavori batch. Il computer eseguiva un lavoro dopo l'altro, seguendo a turno tutti i programmi e svolgendo le relative operazioni di input/output. Man mano che i computer si diffondevano e il costo del tempo macchina si riduceva, i progettisti si misero a cercare modi migliori di impiegare il tempo dei programmatori. Nei primi anni '60 furono implementati i primi sistemi time-sharing, mentre i linguaggi tendevano ancora a essere usati come applicazioni batch. I programmatori volevano lavorare in modo più interattivo, per velocizzare lo sviluppo del software. Volevano un linguaggio che permettesse di collaudare e modificare i programmi in tempo reale. Il BASIC fu la prima risposta a rendersi disponibile.
Progettato per essere un linguaggio utile ai programmatori principianti, il BASIC voleva essere facile da imparare ma potente. E, riuscendoci, diventò il linguaggio principe dei microcomputer. Oggi il BASIC è il linguaggio di programmazione più utilizzato nel mondo.

Amiga Basic

Il BASIC su Amiga ebbe inizi stentati, a causa della apparizione di ABasiC. ABasiC era un'implementazione del linguaggio a dir poco spartana, priva di tutte le comodità del BASIC moderno e ignara delle possibilità di un computer come Amiga. Fortunatamente per la comunità degli utenti ABasiC scomparve velocemente, sostituito da Amiga Basic; un linguaggio che dava ai programmatori ciò che essi chiedevano, accesso alle risorse del computer, con comandi appositi incorporati o chiamate alle librerie software di Amiga.
Soprattutto Amiga Basic, oltre che facile da imparare e piuttosto potente, è stato il linguaggio più accessibile agli utenti Amiga. Di fatto, l'unico linguaggio fornito con Amiga dal 1985.
Se Amiga Basic è incluso in ogni Amiga venduto, come può sperare uno sviluppatore di vendere una nuova versione del BASIC? La risposta è semplice. Amiga Basic è una implementazione meravigliosa, ma non perfetta. Uno dei suoi difetti più evidenti è la mancanza di velocità. L'editor di Amiga Basic è disperatamente lento, capace di passare minuti a trovare una linea ricercata dal programmatore e mancante di numerose opzioni che facilitino la programmazione BASIC. Se però questo fosse l'unico problema, sarebbe anche sopportabile. In realtà, Amiga Basic è lento anche nell'eseguire i programmi. Molto più lento di ABasiC.
E non è finita. Un altro handicap di Amiga Basic, inerente non tanto all'implementazione quanto alle limitazioni intrinseche del BASIC, è la mancanza di supporto di variabili complesse, come i record di Modula-2 o le struct del C. Su una macchina come Amiga, il cui sistema ope-

rativo fa uso esteso di struct, si tratta di una limitazione davvero seria.
Un'altra carenza riguarda la possibilità di avere sottoprogrammi capaci di richiamare se stessi (definiti in gergo programmatorio come ricorsivi). Senza ricorsività, Amiga Basic manca di un elemento importante per la realizzazione di numerosi algoritmi.

Il resto del mondo

Nel mondo del BASIC su Amiga c'è spazio per migliorare e numerosi contendenti pronti ad approfittarne. Purtroppo non sono riuscito a includere nella mia analisi tutti i BASIC per Amiga esistenti, ma l'elenco è sufficientemente ampio per farsi un'idea. Metterò a confronto AC/BASIC 1.3, F-BASIC 2.0, GFA-BASIC 3.0, HiSoft BASIC Professional e True BASIC, oltre ovviamente ad Amiga Basic. La comparazione sarà accompagnata da un'analisi dei singoli BASIC, in cui verranno evidenziati punti di forza e debolezze. AC/BASIC e HiSoft vantano entrambi la compatibilità totale con Amiga Basic. Oltre a maggiore velocità, comprendono la ricorsività e una serie di comandi aggiuntivi. F-BASIC, GFA-BASIC e True BASIC sono invece implementazioni piuttosto differenti. AC/BASIC, F-BASIC, HiSoft BASIC e True BASIC sono compilatori, che possono generare codice autonomo e indipendente, o richiedente librerie di runtime per essere eseguito. GFA-BASIC è invece un interprete.
GFA-BASIC, HiSoft BASIC e True BASIC sono ambienti integrati di sviluppo, ognuno completo di un eccellente editor incorporato, mentre AC/BASIC e F-BASIC prevedono che l'utente scelga un editor di pro-

Tabella 1: I risultati del Benchmark (in ore, minuti e secondi)

	Amiga Basic	AC/BASIC	F-BASIC	GFA-BASIC	HiSoft BASIC	True BASIC
Sieve	1:22/iter.	0:07/iter.	0:00.57/iter.	0:09/iter.	0:07/iter.	0:10.94/iter.
Graphics						
Load IFF	3:07.94	1:13.23	0:02.52(0:57.86)	2:00.45	1:04.76	*
Get and Put	5:34.70	1:51.56	1:02.80	1:29.05	1:49.25	*
Line Drawing	1:59.10	0:53.75	0:27.80	0:14.80	0:53.22	2:09.94
Integer Sort						
Quicksort	$	0:04.46	0:00.36	0:08.13(0:00.01)	0:06.61	0:07.88
Shell sort	05:09.86	0:26.00	0:02.36	0:54.99(0:00.02)	0:23.16	0:10.78
Heapsort	56:03.68	2:33.66	0:00.82	4:38.15	2:06.78	5:00.22
String Sort						
Quicksort	$	01:52.66	0:01.82	0:08.01(0:01.14)	00:18.90	0:08.18
Shell sort	0:05:24.30	08:02.10	0:10.36	0:55.23(0:01.16)	01:07.35	0:11.16
Heapsort	1:02:37.40	52:43.54	0:04.15	5:08.00	12:21.56	5:08.22
Matrix	1:11.88	0:10.52	0:02.72	0:17.0	0:08.42	0:11.64
Floating-point						
Basic Math						
Expected value: 10						
Single-precision						
time	8:05.60	1:30.64	0:54.58	#	1:19.55	%
value	10	10	9.9999999900	#	10	%
Double-precision						
time	10:13.88	2:06.60	3:02.72	1:47.00	2:47.73	1:11.24
value	10	10	10	10	10	10
Area under SIN(x)						
Expected value: 1						
Single-precision						
time	3:51.36	0:58.0	0:19.54	#	0:35.02	%
value	1.000106	0.9998854	0.9999949690	#	1.000106	%
Double-precision						
time	6:11.5	2:44.60	2:30.00	1:27.00	2:47.19	2:41.20
value	1.000000000004508	1.00000000000404	0.9999999998	0.9999999997976	1.000000000002094	1.0
Area under EXP(x)						
Expected value: 1.718281828 (e-1)						
Single-precision						
time	3:48.94	1:09.38	0:18.68	#	0:38.96	%
value	1.718231	1.717434	1.63846	#	1.718231	%
Double-precision						
time	5:23.54	2:49.02	2:32.00	1:25.00	2:44.63	2:55.64
value	1.7182819577210358	1.71828195770946	1.71792146453862	1.718281828586	1.718281957706658	1.71828

* True BASIC doesn't support direct access to the Amiga library routines. Therefore, the load-IFF portion of the graphics benchmark was omitted. (System calls can be written in C or assembly language and linked with the True BASIC program.) True BASIC also doesn't support bitmapped graphics, so the get and put portion was also omitted.
$ Amiga Basic doesn't support recursion, therefore, the quicksort routine was omitted.
\# GFA-BASIC doesn't support single-precision floating-point numbers.
% True BASIC doesn't support single-precision floating-point numbers.

pria preferenza.
Tutti questi programmi, in un modo o nell'altro, sono migliori di Amiga Basic, ma non mancano di pecche. Sta a voi decidere se meritano la spesa di un BASIC aggiuntivo...

I BASIC alla prova

Un'esigenza sempre viva in tutti i programmatori è la velocità. I programmatori cercano sempre di spremere dalle loro macchine il massimo delle prestazioni. Il BASIC non contempla la velocità tra le sue caratteristiche base, ma alcuni di questi pacchetti possono tenere validamente testa a un compilatore C o Modula-2.
Per formulare un giudizio, ho scritto una serie di benchmark basati su quelli di Byte e leggermente rielaborati, per misurare meglio la velocità dei linguaggi senza porre attenzione al processore (che nel nostro caso ovviamente si considera essere sempre lo stesso). I benchmark sono cinque: Sieve (il noto crivello di Eratostene), Graphics (grafica), Sorting (ordinamento), Matrix (calcolo di matrici) e Floating point (operazioni aritmetiche in virgola mobile). I benchmark sono stati eseguiti su un Amiga 2000 con tre Mbyte e processore 68000.

I risultati

Sieve, discendente diretto del test originale presentato su Byte del Giugno 1988, misura la velocità delle operazioni su numeri interi. Il test, che trova numeri primi secondo il sistema del crivello sviluppato dal matematico greco Eratostene, viene eseguito per 50 volte, mostrando il tempo medio impiegato per ogni iterazione. Due dei BASIC hanno ottenuto risultati sorprendenti. F-BASIC ha completato il benchmark con un tempo di 0.57 secondi per iterazione, 12 volte più veloce del secondo classificato. GFA-BASIC ha terminato in nove secondi per iterazione, un tempo appena superiore ai sette secondi di AC/BASIC e HiSoft BASIC. Il tempo di GFA-BASIC è però sorprendente, perché ottenuto da un'interprete, notoriamente più lento di compilatori come sono gli altri programmi.
Graphics consiste di tre parti. Il test esegue il caricamento di un file IFF 320 x 200 a 32 colori, anima una shape 20 x 20 all'interno dell'immagine e poi disegna 10.000 linee casuali sullo schermo. Questo test, più degli altri, sollecita l'interfaccia di ogni linguaggio con le librerie software di Amiga. Il caricamento del file IFF è stato pensato per simulare una situazione tipica di lavoro, presumibilmente frequente su un computer con grandi capacità grafiche. Si può notare dai dati nelle tabelle come il tempo di caricamento da disco sia lo stesso più o meno per tutti i pacchetti messi alla prova. E' la conferma di come la sezione I/O di Amiga sia la "grande equalizzatrice": più se ne fa uso, più si annullano le differenze di prestazioni tra diversi software.
Noterete, anche, che il tempo di F-BASIC è tra parentesi. Il valore riportato è il tempo impiegato dal comando READIFF, proprio del pacchetto, per leggere il file. READIFF è un buon lettore di immagini IFF, ma può essere usato solo con ILBM. Se dovete caricare un'immagine in F-BASIC, READIFF è la soluzione migliore. Se invece dovete caricare un file IFF non ILBM, dovrete scrivere un vostro lettore di IFF.
Non ho implementato la parte di codice per il caricamento di file IFF in True BASIC perché questo linguaggio non supporta le chiamate dirette al sistema operativo di Amiga. In realtà True BASIC consente questo tipo di chiamate, ma bisogna scriverle in C o in Assembler, e poi collegarle al programma BASIC.
La seconda parte di Graphics usa i comandi PUT e GET di Amiga Basic (o gli equivalenti negli altri linguaggi) per muovere una shape sullo schermo, e vuole dare un'idea di come il linguaggio potrebbe comportarsi nella gestione di animazioni. Il benchmark disegna la shape sullo schermo, la cancella e la ridisegna nella nuova posizione per 10.000 volte, restituendo il tempo che appare nelle tabelle.
Nemmeno questa parte di codice è implementata in True BASIC, perché quest'ultimo non supporta la grafica bitmap. Ho tentato di simulare i comandi GET e PUT in vari modi, ma i risultati erano talmente lenti che vi ho rinunciato. True BASIC è ideale per schermi statici, ma l'animazione dovrebbe essere lasciata a linguaggi più adatti al compito.
La terza e ultima parte del benchmark disegna 10.000 linee casuali sullo schermo. Sorprendentemente, GFA-BASIC ha sbaragliato tutti i compilatori in gara, impiegando, ignoro perché, un tempo pari alla metà rispetto al secondo arrivato.
Il test di ordinamento collauda la velocità di un linguaggio nelle operazioni con numeri interi e negli spostamenti di dati nella memoria. Più in dettaglio, ordina 1000 numeri interi scelti a caso per mezzo di vari algoritmi. Per curiosità, ho aggiunto un benchmark per ordinare stringhe di caratteri, che agisce su 1.000 sequenze di 20 lettere ciascuna e utilizza gli stessi algoritmi. In tutte le prove di ordinamento, il migliore si è rivelato di gran lunga F-BASIC.
Guardando le tabelle, noterete una strana anomalia nei tempi di ordinamento di F-BASIC. Nella maggior parte dei casi, l'algoritmo Shell sort dovrebbe essere più veloce dell' heapsort. In F-BASIC non è così, né con i numeri né con le stringhe di caratteri. Siccome tutte le routine avevano a che fare con le stesse sequenze di dati, non si può pensare che Shell sort abbia pagato le conseguenze di un caso particolarmente sfortunato. E anche se proprio questo fosse stato il problema, dubito che entrambe la sequenza di numeri e quella di stringhe fossero casi sfortunati. Ho eseguito la prova numerose volte, cambiando anche le sequenze dei dati, senza trovare una spiegazione o una soluzione al problema.
Dato che Amiga Basic non supporta la ricorsività, non ho implementato l'algoritmo di quicksort. Avrei potuto scrivere un quicksort non ricorsivo, ma sarebbe stato intrinsecamente più veloce del primo; quindi avrei dovuto realizzare in modo non ricor-

sivo anche tutti i quicksort degli altri linguaggi. Invece, ho evitato di scrivere la versione Amiga Basic (sono pigro).
Ancora, consultando le tabelle vedrete che, per GFA, i tempi di quicksort e Shell sort sono tra parentesi. Si tratta delle prestazioni ottenute usando le routine di sort incorporate nel GFA stesso: incredibilmente flessibili, dovrebbe essere possibile utilizzarle per la maggior parte delle applicazioni.
Il benchmark Matrix somma, traspone e moltiplica quattro matrici 20 x 20. Mi aspettavo un successo del True BASIC, ricco di comandi per la manipolazione delle matrici. Come si può vedere, sempre nelle tabelle, anche in questo caso ha avuto la meglio F-BASIC.
Il test per le operazioni in virgola mobile, derivato sempre da Byte di Giugno 1988, verifica la velocità e l'accuratezza del computer con numeri rappresentati in singola e doppia precisione. Il benchmark è diviso in due parti: aritmetica elementare e integrazioni. Nella prima, il computer esegue otto volte un ciclo di 20.000 addizioni di cifre in virgola mobile, ripetendo la procedura per le altre tre operazioni. Tutti i BASIC hanno fornito come risultato i valori previsti, tranne F-BASIC, che in singola precisione ha sbagliato dello 0.0000001%, a dimostrazione di una ricerca della velocità a scapito della precisione. Se volete programmi superveloci e non avete grandi problemi di accuratezza, F-BASIC è ideale, e costa molto meno di un processore matematico aggiuntivo. Se il vostro Amiga deve far decollare uno Space Shuttle, invece, vi consiglio caldamente il ricorso a un altro BASIC.
La seconda parte del benchmark calcola l'area di due funzioni seguendo il metodo di integrazione della regola del trapezio. Come potevate aspettarvi, i risultati sono molto meno omogenei di quelli ottenuti con le quattro operazioni.
Né GFA-BASIC né True BASIC supportano la singola precisione, così le loro prestazioni sono misurate solo in doppia precisione. F-BASIC ha ottenuto i migliori tempi e una buona precisione nella funzione SIN, bilanciata da uno strano, brutto risultato con EXP. True BASIC è stato il più preciso su SIN, e GFA su EXP. Nessuno dei pacchetti si è rivelato orrendamente impreciso, e solo F-BASIC si è distinto per le sue doti di sprinter. Globalmente, il test in virgola mobile non ha fornito un vero vincitore, presentando tutti i programmi su un livello equivalente e comunque soddisfacente per l'utente medio.

Tabella 2: Lunghezza dei file (in byte)

	AC/BASIC	F-BASIC	HiSoft BASIC	True BASIC
Sieve				
Library	3,300	35,108	1,468	692
Stand-alone	46,612	60,032	*	77,780
Graphics				
Library	11,160	6,176	6,920	$
Stand-alone	54,472	31,376	*	$
Integer Sort				
Library	6,612	12,560	3,216	1,991
Stand-alone	49,924	37,780	*	79,080
String Sort				
Library	6,720	5,156	3,472	1,996
Stand-alone	50,032	30,768	*	79,072
Matrix				
Library	4,980	8,916	2,548	875
Stand-alone	48,292	33,932	*	78,000
Floating-point				
Single-precision				
Library	7,320	7,056	3,776	%
Stand-alone	50,632	33,836	*	%
Double-precision				
Library	7,904	7,520	4,144	1,916
Stand-alone	51,216	31,504	*	79,152
Runtime Library	43,264	39,812	46,672	76,852

* The version of the HiSoft compiler used to conduct these tests would not generate stand-alone programs. This bug has been corrected in the latest release.
$ The True BASIC version of the graphics benchmark contained only one of the three sections. Therefore, the file sizes would be meaningless.
% True BASIC doesn't support single-precision floating-point numbers.

Programmi per taglie forti

I compilatori BASIC tendono, in generale, a generare programmi dalle dimensioni a dir poco generose, in quanto quasi tutti accludono al codice un modulo runtime. In effetti, quando si usa un interprete BASIC è il linguaggio stesso ad assumere il ruolo di modulo runtime, e i moduli acclusi ai programmi dai compilatori sono non dissimili da interpreti privi di editor. Se il compilatore includesse nel codice solo le routine realmente usate, i programmi sarebbero molto più piccoli, come accade in linguaggi tipo C o Modula-2.
I quattro compilatori qui presentati possono generare programmi che richiedono moduli runtime, oppure programmi autosufficienti. I primi sono di dimensioni ridotte rispetto ai secondi, ma vogliono comunque il pacchetto di runtime per girare. Se avete più programmi BASIC compilati sullo stesso disco, può essere economicamente vantaggioso (dal punto di vista dell'occupazione di

Tabella 3: Tempo medio di compilazione (in secondi)

AC/BASIC	F-BASIC	HiSoft BASIC	True BASIC
57.34	57.75	59.10	89.27

spazio) tenere un solo modulo di runtime sul disco, condiviso da tutti i programmi.
L'unico problema è che dovete ricordarvi, se trasferite un programma su un altro disco, di farlo accompagnare da una copia del runtime. Un secondo problema, più serio, è far sì che il programma sappia dove si trova effettivamente il runtime. Così, sebbene i programmi autosufficienti siano più grossi degli altri, sono senz'altro più facili da gestire e, per esempio, condividere con altri utenti. Nelle tabelle, potete vedere le dimensioni dei programmi realizzati per i benchmark, con i rispettivi runtime.

Compilatore più veloce

Ho misurato la velocità dei compilatori eseguendo un collaudo con una versione di "The Hermit", un adventure game testuale lungo, nella versione per Amiga Basic, 581 linee di codice.
I risultati sono discreti, con True BASIC che scivola verso la fascia della lentezza. I tempi dati nella tabella si riferiscono a programmi runtime-dipendenti; le versioni autosufficienti, con alcuni dei compilatori (AC/BASIC e HiSoft BASIC), impiegano un tempo leggermente superiore.
AC/BASIC e HiSoft BASIC possiedono entrambi opzioni software che permettono di generare programmi indipendenti o legati al runtime.
F-BASIC e True BASIC usano invece un programma specifico che, se desiderato, collega il runtime al software realizzato rendendolo autonomo. A questo riguardo, il Link di F-BASIC è velocissimo, e impiega solo qualche secondo in più. Il Binder di True BASIC, invece, può impiegare anche interi minuti. Fortunatamente, l'operazione di collegamento del runtime si esegue, in genere, una sola volta. Se si dovesse usare il Binder tutte le volte che si vuole collaudare un programma, il compilatore sarebbe virtualmente inutilizzabile.

I BASIC al microscopio

AC/BASIC è un compilatore a passo multiplo che vanta una completa compatibilità con Amiga Basic. Ha un'interfaccia a menu, facile e intuitiva da usare, e può essere lanciato sia da CLI che da Workbench.
L'unico difetto dell'interfaccia di AC/BASIC è, a mio parere, la mancanza di un file requester. AC/BASIC si è rivelato il più compatibile dei linguaggi qui presentati rispetto ad Amiga Basic, presentando problemi solamente su due programmi tra i 70 compilati per il test.
Entrambi presentavano errori di sintassi in zone di codice apparentemente mai soggette a esecuzione. I programmi generati da AC/BASIC si sono dimostrati molto solidi, quasi mai fonte di Guru Meditation. I comandi BASIC che danno i maggiori problemi al pacchetto sono, comunque, quelli per la sintesi vocale e sonora. I suoni vengono eseguiti troppo in fretta, o non eseguiti del tutto. La voce, quando funziona, funziona bene, altrimenti provoca una Guru. Il manuale di AC/BASIC è bene organizzato e sufficientemente completo. Gli autori hanno visibilmente impiegato più tempo a spiegare come funziona il compilatore che non a riscrivere il manuale di Amiga Basic.
Complessivamente, AC/BASIC merita un voto medio. Se cercate la velocità potete trovare di meglio, ma non se volete la compatibilità con Amiga Basic. Sospetto che quest'ultimo fosse l'obbiettivo dei programmatori che hanno realizzato AC/BASIC, e devo dire che lo hanno raggiunto brillantemente.

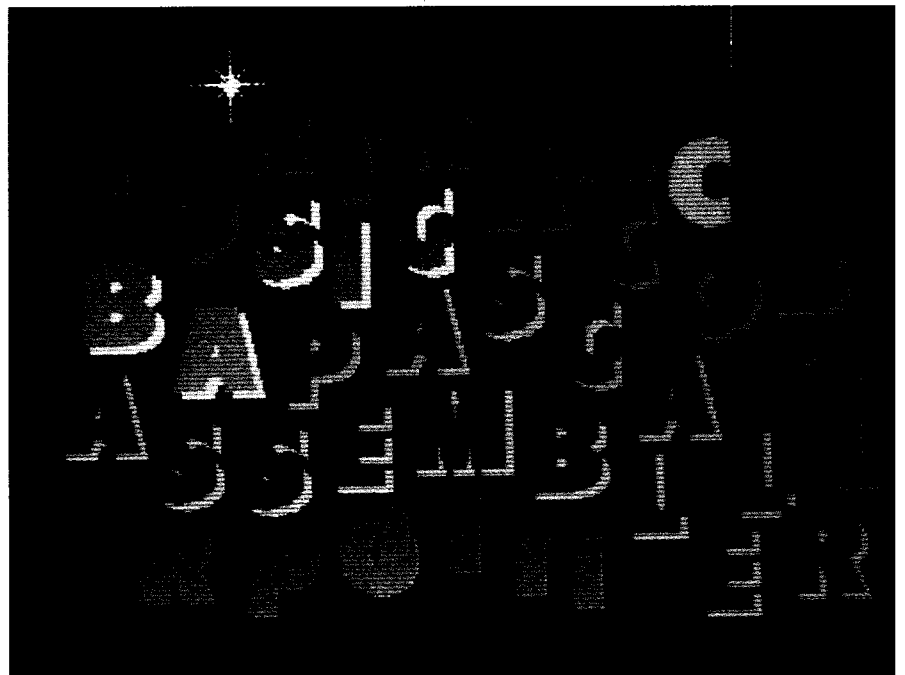
F-BASIC

F-BASIC è un compilatore BASIC a passo singolo progettato con uno scopo: la velocità (F deve stare per Fast, veloce!). Accetta in input file di testo ASCII e genera in uscita programmi in codice macchina per il 68000. F-BASIC effettua numerose ottimizzazioni, razionalizzando tra l'altro l'uso degli operatori logici AND e OR e quello delle variabili. F-BASIC ha battuto i suoi concorrenti quasi in tutti i benchmark.
Per darvi un'idea della sua velocità, sappiate che il compilatore stesso è stato scritto in F-BASIC.
Il disco del programma include il compilatore, un linker (per attaccare a un programma i moduli di runtime), numerose librerie e un debugger simbolico. Tutti i programmi del pacchetto devono essere usati da CLI, così come quelli generati da FastCom.
Oltre alla velocità, F-BASIC possiede altre due grandi doti che lo distinguono dagli altri BASIC: i record (chiamati struct in alcuni linguaggi) e il debugger simbolico.
Su una macchina come Amiga, scrivere un programma funzionale senza record è come estrarre un dente del giudizio senza anestesia: doloroso! F-BASIC è l'unico BASIC per Amiga a supportare variabili complesse, veramente un piccolo capolavoro della Delphi Noetic. Se volete programmare seriamente su Amiga, senza dover imparare C o Modula-2, F-BASIC è il linguaggio per voi. Il debugger, SLDB, può ridurre del 50% il vostro tempo di revisione e correzione. Potete visualizzare codice sorgente, codice macchina, variabili, record o i registri del processore. Potete piazzare breakpoint nel codice a vostro piacere, comprese interruzioni temporizzate, che scattano se attivate da un contatore. Potete far funzionare il codice finché non si incontra un breakpoint, o eseguirlo linea per linea. SLDB permette perfino di alterare le variabili, i registri o le locazio-

ni di memoria durante l'esecuzione di un programma. SLDB è uno strumento di programmazione di grande valore, di quelli di cui ci si chiede come se ne faceva a meno, prima di averlo.
Non voglio far pensare, comunque, che F-BASIC sia il compilatore BASIC perfetto; anche lui ha le sue "magagne". In un certo senso, F-BASIC assomiglia più al Pascal che al BASIC. Richiede uno statement PROGRAM all'inizio di ogni programma: tutte le variabili vanno dichiarate prima di usarle e le stringhe sono implementate come vettori (array) di caratteri. Queste differenze tra F-BASIC e Amiga Basic possono trasformarsi in un muro, per chi non vuole imparare una sintassi per molti versi nuova.
Uno dei problemi principali che ho avuto con F-BASIC è stato la lettura di stringhe da un file su disco. Con qualunque tipo di routine, era impossibile leggere più di 7000 caratteri da un file di testo. La lettura avveniva perfettamente e, dopo i 7000 caratteri, arrivava una Guru Meditation con numeri apparentemente casuali. Nell'impossibilità di spiegare e risolvere la cosa, per eseguire il test di sort ho frammentato il file di numeri interi e quello di stringhe in file più piccoli.
F-BASIC comprende un manuale d'uso, una guida all'utilizzo del debugger a livello di codice sorgente e un manuale di aggiornamento, versione 2.0, più disco di sistema, disco di upgrade del sistema e disco con programmi dimostrativi. Sebbene completi di ottimi indici, i manuali si sono rivelati incompleti e difficili da usare. Parte del problema sta nel fatto che alcune informazioni sono semplicemente assenti, parte consiste nell'estrema attenzione necessaria per girare le pagine dei manuali senza staccarle.
C'è da dire qualcosa anche sull'impaginazione. L'aspetto dei manuali è simile a quello di fogli A4 scritti a macchina, con pochissimi spazi bianchi e nessun cambio di carattere, eccezion fatta per le sottolineature che accompagnano l'inizio di ogni sezione. Di solito si trova ciò che si sta cercando, ma bisogna avere pazienza. Nonostante tutto, la mia valutazione di F-BASIC è ottima. Anche se i manuali sono difficili da leggere e il Workbench non è supportato, il programma mi è piaciuto ugualmente, quasi contro la mia volontà. C'è qualcosa di intrigante in un compilatore che può correre velocissimo mentre tutti gli altri trascinano i piedi.

GFA-BASIC

GFA-BASIC è l'unico interprete qui esaminato. Scritto in Germania Occidentale, GFA è un nuovo linguaggio, quasi del tutto incompatibile con Amiga Basic. In compenso, è compatibile con le versioni 2.0 e 3.0 del GFA-BASIC su Atari ST, dal quale ho prelevato molti programmi, a cui bastava solo convertire opportunamente le chiamate al sistema operativo per ritrovarseli funzionanti in pochissimo tempo.
Avendolo già provato su ST, GFA-BASIC è risultato estremamente familiare su Amiga. L'editor sulle due macchine è quasi identico, e contiene funzioni (correzione automatica di maiuscoli e minuscoli, indentazione automatica, una funzione molto veloce di ricerca e sostituzione) che alleviano la fatica del programmatore. Ci sono anche cose un pochino frustranti, per esempio l'impossibilità di spostare il cursore da una linea fino a che non è sintatticamente corretta.
Per un interprete, GFA-BASIC è incredibilmente veloce. Nella maggior parte dei benchmark, riesce a ben figurare nel confronto con i compilatori. Se la GFA Systemtechnik, autrice del programma, mantiene le sue promesse, dovremmo vedere pronto ben presto un compilatore GFA. E se quest'ultimo dovesse essere veloce come l'interprete, non c'è dubbio che F-BASIC si troverebbe di fronte a un rivale agguerrito. GFA-BASIC supporta molte delle capacità di Amiga con comandi interni, ma il suo lato più bello è l'interfaccia verso le librerie di sistema del computer.
La Systemtechnik ha incluso nella confezione un programma che converte i file standard Commodore FD in subroutine GFA. Per chiamare una routine di libreria dal programma, basta dimensionare un array chiamato m68%, e includere nel software che si sta scrivendo la corrispondente subroutine.
GFA-BASIC supporta molti comandi di basso livello, che non ci aspet-

terebbe normalmente in un BASIC. Istruzioni come BCLR, BSET, BCHG, ROL e ROR rendono la programmazione a livello di bit non solo possibile, ma anche facile. Si tratta di possibilità di cui uno non sente la mancanza, ma a cui non si può più rinunciare una volta che ci sono.
Anche con tutti i suoi lati positivi, GFA-BASIC non supporta le variabili complesse (record e struct). Senza di esse, è un meraviglioso linguaggio per la programmazione. Se le avesse, sarebbe un sogno fatto realtà. Il voto finale per GFA-BASIC può essere solo eccellente. Dei BASIC che ho usato su Amiga, GFA è quello che preferisco e, se Systemtechnik darà alla luce il sospirato compilatore, diventerà l'ambiente BASIC definitivo per programmare su Amiga. (Beh, quasi, visto che non ci sono le struct). Si potrebbero sviluppare i programmi sull'interprete, velocissimo di per sé, e poi creare un fantastico prodotto finito con il compilatore.

HiSoft BASIC

HiSoft BASIC è un compilatore a passo multiplo compatibile con Amiga Basic. Il sistema include un editor integrato, e tutto può essere richiamato indifferentemente da CLI o da Workbench.
L'editor di HiSoft BASIC mi ricorda quello di Amiga Basic. I tasti di controllo del cursore sono simili, e lo stesso si può dire dei metodi per evidenziare il testo ed eseguire le operazioni di taglia-e-incolla. Una differenza rilevante, invece, è che l'editor HiSoft è veloce. L'unico difetto che ho trovato è che il dialog box per richiedere i file non si ricorda della directory utilizzata in precedenza dall'utente, come accade su altri sistemi.
HiSoft BASIC afferma di essere compatibile con Amiga Basic; nonostante ciò ho trovato difficoltà a far compilare programmi scritti con quest'ultimo, anche se generalmente gli interventi da eseguire non andavano oltre la riscrittura di qualche linea e qualche cambio di sintassi. Il codice generato da HiSoft tende a essere meno ingombrante di quello di AC/BASIC. Anche le differenze di velocità sono minime, eccetto quando si fa uso di stringhe. Apparentemente, HiSoft BASIC fa uso di stringhe statiche per velocizzare le operazioni di gestione del testo. Come potete vedere dai test, la superiorità di HiSoft in questo campo è nell'ordine del 400%-700% rispetto ad AC/BASIC.
Come AC/BASIC, HiSoft ha problemi nel compilare programmi contenenti suono o sintesi vocale, problemi anche più gravi, visto che non si riesce a evitare la comparsa della Guru in nessun modo.
I programmi "muti", invece, sono stabili come franchi svizzeri sul mercato valutario... HiSoft è andato un po' in crisi anche con i programmi che usavano GET e PUT per muovere una shape sullo schermo, ma sospetto che questo inconveniente sia stato risolto nell'ultimissima versione del pacchetto. Una cosa che realmente mi piace di HiSoft sono le estensioni ad Amiga Basic.
Il programma contiene fra l'altro statement SELECT...CASE, INC(rement) e DEC(rement) e REPEAT...UNTIL, che contribuiscono non poco a facilitare la programmazione (anche quella dei benchmark). Il manuale è completo e strutturato molto bene. Comincia con un'introduzione al compilatore, e prosegue con un eccellente tutorial, seguito da esaurienti spiegazioni sull'uso del compilatore. Il resto del libro è dedicato all'elenco dei comandi e agli indici di riferimento. Ogni comando viene descritto nei dettagli, con esempi di uso. Il voto di HiSoft BASIC è sopra la media. Sebbene le sue prestazioni siano pressoché identiche a quelle di AC/BASIC, i comandi aggiuntivi e un eccellente manuale lo pongono a un livello più alto del suo rivale più immediato.

True BASIC

True BASIC è un compilatore realizzato con una filosofia radicalmente differente da quella degli altri linguaggi fin qui presentati. True BASIC è stato realizzato dai creatori del linguaggio stesso. La loro meta era l'implementazione di un BASIC veramente standard, che non desse problemi di compatibilità ma, soprattutto, fosse identico su ogni macchina, così che il software potesse funzionare indifferentemente dall'hardware. Questo presupposto rende molto difficile una reale comparazione con gli altri linguaggi. E' un po' come la vecchia operazione tra mele e pere... Forse può essere utile cercare di stabilire quale sia il mercato per questo genere di prodotto. Come programmatore, non userei mai un linguaggio incapace di approfittare delle capacità dei vari sistemi operativi supportati dai rispettivi hardware. Immagino che True BASIC possa essere particolarmente utile a livello scolastico, ma la mia impressione rimane sfavorevole.
True BASIC comprende un manuale di riferimento, una guida per l'utente e il disco di sistema, contenente il compilatore e svariate dozzine di programmi dimostrativi. In aggiunta al pacchetto standard ho ricevuto il modulo runtime, completo di dischetto e un piccolo manuale.
La guida all'utente presenta True BASIC in un modo un po' prosaico, che piacerà al programmatore neofita, e fornisce alcune informazioni specifiche riguardanti l'uso su Amiga. Il manuale di riferimento descrive il linguaggio e passa a presentare gruppi di comandi in relazione tra loro. Entrambi i manuali sono ben scritti, al punto che li considero i migliori manuali per computer che ho visto.
True BASIC è molto simile al BASIC standard (gli autori lo presentano come il BASIC standard), e molto diverso al tempo stesso.
Ricalca sufficientemente il BASIC standard per consentire la stesura veloce di programmi, e se ne distacca quel tanto che basta per farvi impazzire durante il debugging. Per esempio, volendo scrivere una stringa in un file, con Amiga Basic si digita:

OPEN "nomefile" FOR OUTPUT AS

1:PRINT #1, "Questa è una stringa": CLOSE #1

Guardate l'equivalente in True BASIC:

OPEN #1: NAME "nomefile", ACCESS OUTPUT, CREATE NEWOLD, ORGANIZATION TEXT; PRINT #1:"Questa è una stringa"; CLOSE #1

Si noti che la separazione di due comandi è espressa con il punto e virgola, e i due punti vengono usati, invece, all'interno delle istruzioni. La sintassi da comando a comando tende ad assomigliarsi, ma se siete abituati a un altro BASIC programmare in True BASIC diventa un'esperienza interessante. Molte istruzioni, poi, non condividono nemmeno il nome con la loro controparte in BASIC standard.

Come GFA-BASIC, anche True BASIC possiede comandi che non ci si aspetterebbe di trovare in un BASIC comune, ma in questo caso essi sono di livello molto alto. Per esempio, True BASIC ha istruzioni per leggere matrici da disco o da statement DATA; per invertire, moltiplicare e in genere operare su matrici; per riunire primitive grafiche in funzioni richiamabili con il comando DRAW. Si può dire che in generale True BASIC tenda a nascondere il computer all'utente e al programmatore, a renderlo in qualche modo trasparente. Come già detto, scrivere programmi True BASIC specifici per un dato computer è un'impresa quasi impossibile.
La dote più affascinante di True BASIC è la capacità di ottenere l'output grafico nella scala desiderata. Se il rapporto tra dimensione orizzontale e verticale del vostro computer è anomalo, basta un comando per essere sicuri che tutte le istruzioni grafiche, da quel momento, si adatteranno alla configurazione indicata. Probabilmente è questa la ragione per cui i comandi grafici di True BASIC tendono a essere un po' più lenti rispetto agli altri BASIC. Se dovete disegnare grafica statica, come un istogramma o il tracciato di una funzione matematica, True BASIC è ottimo. Scordatevi, invece, l'animazione.
Il voto finale per True BASIC è tutto sommato scarso. Il linguaggio non può fare uso delle capacità uniche di Amiga senza ricorrere a routine realizzate in un altro linguaggio. Questo punto può apparire secondario, ma rivela un dato di fatto importante: True BASIC è un eccellente linguaggio teorico ma, come il Pascal, è poco utilizzabile in molti tipi di applicazione. ▲

Professional Draw: l'altro modo di disegnare

Professional Draw è il primo programma di grafica strutturata per Amiga che permette di realizzare ottimi disegni.

Daniele Cassanelli

La Gold Disk, software house produttrice di Professional Draw, è famosa già da qualche anno per la produzione del primo programma di DeskTop Publishing per Amiga, PageSetter, che mostrò subito al pubblico le potenzialità della macchina in questo campo. Successivamente la Gold Disk commercializzò un altro programma di DTP, Professional Page che, oggi alla release 1.3, si avvicina seriamente a programmi professionali come Ventura, ambiente MS-DOS, e Page Maker, per Macintosh (a titolo informativo, esiste anche Page Maker per ambiente MS-DOS).
E' proprio dal mondo Apple che la Gold Disk prende spunto per commercializzare un programma simile a FreeHand e Illustrator, programmi basati cioè sulla grafica strutturata.

1

La grafica strutturata

Differentemente dalla grafica a cui siamo abituati (bitmap graphics, quella di Deluxe Paint per intenderci) dove un'immagine viene descritta attraverso migliaia di pixel di colore diverso, la grafica strutturata è basata su elementi grafici (come linee, curve, ellissi e rettangoli) e produce una descrizione dell'immagine piuttosto che l'immagine stessa. La principale differenza tra un programma come DPaint e Professional Draw è che, una volta disegnato un oggetto, ad esempio un cerchio, esso risulterà sempre modificabile grazie ai suoi "punti di controllo" che ne delimitano la forma e la grandezza. Per avere un'idea di quanto detto sopra, guardiamo la figura 1: il secondo cerchio, ottenuto tramite le classiche funzioni di copia e incolla (copy e paste), è una copia fedele del primo ed è stato semplicemente modificato spostando un suo punto di controllo ottenendo così una forma completamente diversa. Tutti gli oggetti in Professional Draw possono essere modificati in questa maniera, dandoci così la possibilità di cambiare i nostri lavori in un batter d'occhio.

Il programma

Professional Draw viene fornito su due dischi, Program Disk e Fonts & Utilities Disk, in un'elegante confezione sulla quale vi è rappresentato il "Canone delle proporzioni", il famosissimo disegno di Leonardo Da Vinci indicante la perfezione.
Nella confezione troviamo, oltre ai suddetti dischi, il manuale del programma (130 pagine, in inglese,

automaticamente un cerchio, da un rettangolo passeremo ad un quadrato, mentre nel caso di linea retta essa verrà tracciata con un'angolazione multipla di 45 gradi.
Il penultimo tool grafico permette di disegnare una griglia rettangolare (quadrata se tracciata premendo Alt). Il numero di linee presenti all'interno della griglia si può determinare cliccando due volte sul gadget in questione; immediatamente apparirà una finestra di dialogo (dialog box) dove viene chiesta l'immissione del numero di linee orizzontali e verticali desiderate. In generale la finestra di dialogo è implementata in quelle funzioni che richiedono precisione nell'uso (ad esempio nella rotazione di un oggetto). Infine, l'ultimo tool (freehand tool) permette di tracciare linee a mano libera.

chiaro e ben documentato) e i classici documenti da spedire alla software house per la registrazione come nuovo utente. Professional Draw è in grado di lavorare sia in bianco e nero sia a colori e richiede per il funzionamento minimo un Megabyte. Diretto discendente di Professional Page, eredita da questo l'impostazione generale: Professional Draw, infatti, è un programma "page oriented" (ossia tutto il lavoro è organizzato su pagine, di cui possiamo scegliere dimensioni e margini), inoltre ha mantenuto alcuni menu identici al "padre impaginatore", questo per non lasciare disorientato l'utente nel passaggio da un pacchetto all'altro. Sulla destra della pagina di lavoro troviamo una doppia fila di gadget, chiamata tool palette, indispensabili per l'uso di Professional Draw. In cima a questa "palette di strumenti", sotto i gadget di profondità dello schermo, abbiamo quello per cambiare pagina di lavoro (sia sequenzialmente, tramite frecce, sia saltando ad una determinata pagina). Immediatamente sotto ve ne sono altri quattro utili per il funzionamento generale di Professional Draw: troviamo il "null pointer", per la selezione di un oggetto, il "marquee tool", per la selezione contemporanea di più oggetti, il "magnify tool" per l'ingrandimento della pagina e quello per il suo spostamento. Sotto questi gadget ne troviamo uno più grosso di posizionamento della pagina (che rimpiazza con successo le barre di scorrimento laterali), molto utile quando si lavora in ingrandimento su di un particolare, anche se risulta poco maneggevole nei forti ingrandimenti. Successivamente troviamo sei gadget per i tool grafici (drawing tools) e altri sei per la modifica degli oggetti (editing tools).

I tool grafici

Come dicevamo all'inizio, Professional Draw è basato su oggetti grafici fondamentali. Essi sono tutti disponibili sulla destra della pagina di lavoro. Ma andiamo con ordine: innanzitutto troviamo il gadget per tracciare linee rette, a fianco del quale si trova quello per l'inserimento dei testi, argomento di cui parleremo più avanti. Altri gadget presenti sono quelli per tracciare ellissi e rettangoli. Un'utilissima funzione è svolta dal tasto Alt: esso "costringe" un oggetto grafico ad essere regolare; nel caso dell'ellisse, ad esempio, premendo il tasto Alt otterremo

Modifica degli oggetti

Una volta creati gli oggetti è possibile modificarne la forma grazie ad alcuni tool presenti sotto quelli grafici appena visti. Dopo aver selezionato uno o più oggetti (tramite il tasto Shift, esattamente come nella selezione delle icone del Workbench) è possibile ridimensionarli a nostro piacimento, sia in senso orizzontale e/o verticale, sia in modo "regolare", mantenendo cioè le proporzioni originali (tramite Alt); possiamo poi ruotare gli oggetti selezionati (cliccando due volte sul gadget è possibile immettere l'angolo di rotazione, mentre tenendo premuto il tasto Alt lo ruotiamo di 45 gradi per volta), o ancora ottenere gli oggetti riflessi rispetto ad un asse e, tramite il distortion tool, distorcerli a nostro piacimento. Gli ultimi due tool di manipolazione degli oggetti sono chiamati "hook" e "cutting" che servono rispettivamente per aggiungere un punto ad un oggetto (per un miglior controllo della forma) e per troncare in un punto di controllo un determinato oggetto.

Le curve di Bezier

Tutte le linee tracciate in Professional Draw sono particolari curve det-

te di Bezier, particolarmente utilizzate nella grafica strutturata date le loro caratteristiche.
Le curve di Bezier presentano, in generale, solo quattro punti di controllo: due di questi punti, detti di ancoraggio, determinano l'inizio e la fine della curva stessa; gli altri due punti detti di direzione o di tangenza, determinano la forma della curva. Cliccando, ad esempio, su di un punto di ancoraggio, vedremo comparire un'ulteriore retta che rappresenta esattamente la tangente geometrica della linea nel punto di ancoraggio. Muovendo i punti di controllo della tangente, vedremo mutare simultaneamente la forma della linea, strumento molto utile nei piccoli ritocchi.

L'inserimento del testo

Abbiamo visto nei "drawing tool" che è possibile inserire del testo nella nostra pagina di lavoro.
L'aspetto affascinante di questa possibilità è che anche i caratteri caricati sono oggetti e possono, quindi, essere ridimensionati, ruotati e distorti come un qualsiasi altro oggetto. Naturalmente non possono essere caricati i normali font, ma solo quelli forniti nel secondo disco (Fonts & Utilities Disk) riconoscibili per l'estensione ".pdfont". Dopo aver cliccato nel punto in cui si vuol far apparire il testo, si apre una finestra dove viene scelto il tipo di carattere, la dimensione e la frase da immettere. Sono disponibili due tipi di font, Times e Univers (simile all'Helvetica), in otto dimensioni (da 24 a 144 punti).

Le opzioni dai menu

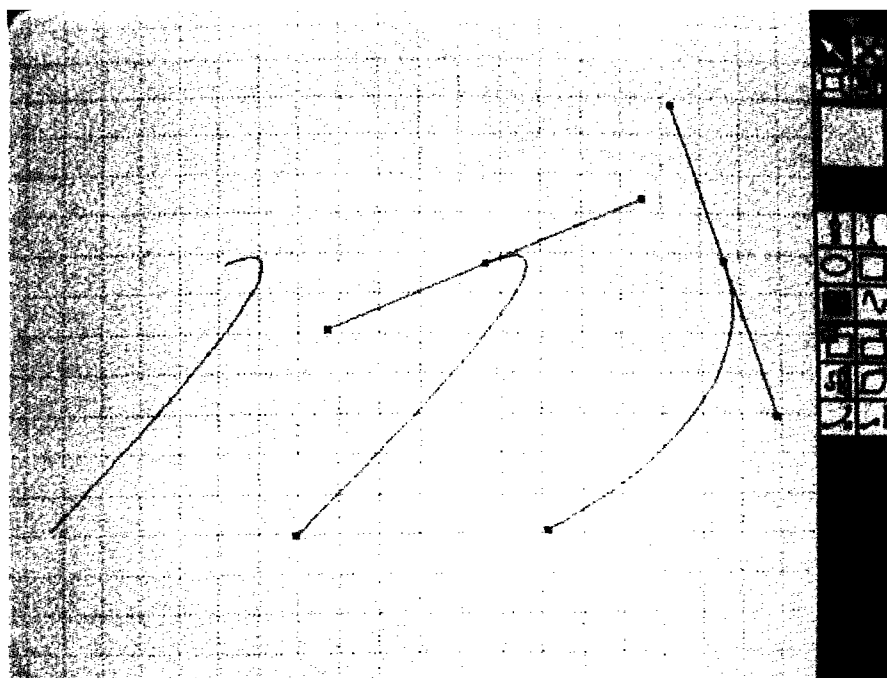
Non tutte le operazioni in Professional Draw possono essere svolte dal "tool palette".
Dai menu a tendina, infatti, vengono svolte operazioni meno frequenti. Dal menu Object, ad esempio, è possibile "raggruppare" più oggetti in uno solo, eseguire l'operazione contraria, bloccare e sbloccare gli oggetti. Inoltre, è possibile modificare lo spessore della linea, il pattern (oltre a quelli predefiniti, è possibile crearne uno personalizzato), il colore e stabilire se l'oggetto deve essere riempito.
Professional Draw, inoltre, permette di salvare gli oggetti singolarmente dandoci così la possibilità di creare librerie personalizzate chiamate clip.
Per finire dal menu Preferences possiamo scegliere l'unità di misura dei righelli, se Professional Draw deve lavorare in Interlace o no e se deve utilizzare il WYSIWYG per visualizzare gli oggetti (se non viene scelta questa opzione gli oggetti vengono visualizzati in wire-frame)

Stampa Postscript e EPSF

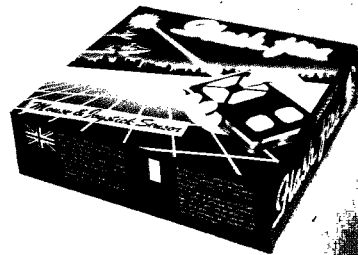
Una delle maggiori peculiarità di Professional Draw è la possibilità di salvare o stampare in Postscript, ossia di poter utilizzare stampanti laser per ottenere un migliore risultato su carta. E' possibile, inoltre, salvare il lavoro anche in EPSF (Encapsulated Postscript Format).
Entrambi i formati contengono una descrizione dell'immagine in linguaggio Postscript; il formato EPSF contiene in più una descrizione del contenuto del file, in modo da essere importato in quei programmi d'impaginazione che supportano il suddetto formato. In questo modo è possibile, ad esempio, importare un disegno prodotto con Professional Draw in Professional Page o, se necessario, in PageMaker e Ventura.
Questo, comunque, non è l'unico modo per importare i lavori in Professional Page, dato che in questo programma esiste una specifica opzione per il caricamento degli oggetti di Professional Draw.

Conclusioni

Professional Draw è dunque un ottimo programma di grafica, ed essendo il solo nel suo campo non è possibile effettuare un confronto con altri pacchetti del genere; un confronto con Illustrator o Freehand del Macintosh è tuttavia inutile, viste le grandi differenze tra le due macchine. Il prezzo di Professional Draw è di circa 450.000 lire, tutto sommato adeguato al livello del pacchetto. ▲

**Professional Draw
è in vendita presso:
Pix Computer s.r.l.
Via F. D'Ovidio, 6c - 00137 Roma
Tel 06/8293507 - 825731**

GIOCARE AI BARBARIAN PRENDENDO IL THE?

Veloce, preciso, comodo e sensibile come nessun altro, ***FlashFire Mouse & Joystick Sensor*** funziona su tecniche d'induzione capacitiva. I suoi tasti sensoriali sono in grado di trasmettere i comandi per mezzo del semplice sfioramento, con velocità e precisione mai registrate prima d'ora, e senza i problemi di avaria alle parti meccaniche dei joystick tradizionali. ***FlashFire Mouse & Joystick Sensor*** unisce tecnologia elettronica d'avanguardia ad un design raffinato e funzionale. Particolarmente indicato per coloro che non si accontentano delle prestazioni normalmente raggiungibili nei Games, ***FlashFire*** è anche un mouse avanzato che elimina i fastidiosi problemi di spazio d'azione sulla scrivania, oppure, grazie alla precisione e alle prestazioni ottimali nelle applicazioni di computer graphics. Provate subito la sensazione di prendere il futuro nelle vostre mani.

FlashFire è in vendita nei migliori negozi.

È prodotto da:

GP ELETTRONICA

ITALIAN APPLY ELECTRONIC SY

Via IV Novembre 32/34 - 20092 Cinisello Bal
Telefono 02/6189551 - Facsimile 02/660

Programmi Editor/Librarian nella Workstation MIDI

Aldo & Andrea Laus

Uno dei modi più interessanti per utilizzare un computer, all'interno del mondo MIDI, è l'applicazione Voice Editor/ Librarian, con cui è possibile creare nuove voci sul vostro strumento MIDI (funzione Editor), e salvarle, sotto forma di file, su dischetti (funzione Librarian), realizzando una vera e propria libreria di suoni, pronti per essere caricati sullo strumento al momento opportuno.
I moderni sintetizzatori e i relativi moduli expander, sono generalmente dotati di potentissimi sistemi di generazione di timbri, la cui complessità di gestione è però tale che i costruttori rendono già disponibile un gran numero di timbri, pronti all'uso, chiamati preset.
Di solito però, questi stessi strumenti consentono all'utente di creare nuovi suoni e di immagazzinarli in altrettante memorie RAM, già presenti sullo strumento.
Questi suoni possono poi anche essere salvati o direttamente su cartuccia RAM esterna, o su unità a disco, presenti sullo strumento, o nella memoria di un computer esterno, tramite il Sistema Esclusivo MIDI dello strumento. La creazione (edit) di suoni non è assolutamente facile, soprattutto se effettuata direttamente sullo strumento, poiché avendo come unico riferimento il piccolo display LCD, di cui sono dotati tutti gli strumenti, si ha solo una visione parziale dell'editing e dopo un po' di tentativi, lo diciamo per esperienza, ci si stanca e si abbandona la prospettiva di creare nuovi suoni.
Era sicuramente più facile editare i suoni sul vecchio MiniMoog, in cui tutte le manopole erano in vista e, quindi, le loro posizioni relative erano facilmente gestibili. Alcuni costruttori di strumenti, sensibili alle esigenze della clientela, hanno realizzato dei dispositivi hardware, denominandoli "programmatori di voci" (vedi foto 1) che, in effetti, presentano una serie di potenziometri per mettere a proprio agio i suonatori. In alternativa a questa pratica ma costosa soluzione, ci vengono in aiuto, oggi, i programmi editor che presentano sul video tutti i parametri che possiamo comodamente variare servendoci del mouse o della tastiera del computer.
Ovviamente, a parte le questioni di prezzo, riteniamo più versatile e comodo l'utilizzo di programmi editor, specialmente se abbiamo più di uno strumento su cui agire.
Il mercato offre già diverse alternative di editor per i più noti strumenti musicali in commercio.
In alcuni casi, grazie ad esempio alla funzione multitasking di Amiga, è anche possibile il funzionamento in contemporanea di un programma sequencer e di un editor/librarian; ciò consente, ad esempio, di verificare gli effetti delle variazioni dei suoni mentre viene eseguito il brano! Generalmente, i programmi voice editor sono dedicati ad un preciso strumento e difficilmente si possono adattare ad altre macchine, in quanto ogni strumento è diverso dagli altri (salvo rari casi di strumenti appartenenti alla stessa famiglia).

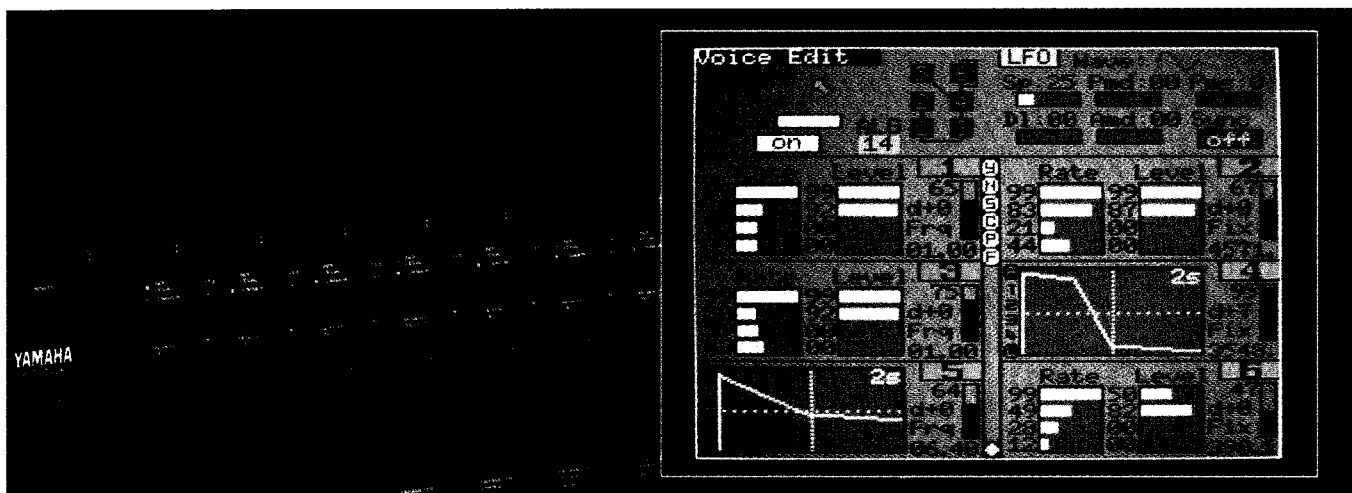
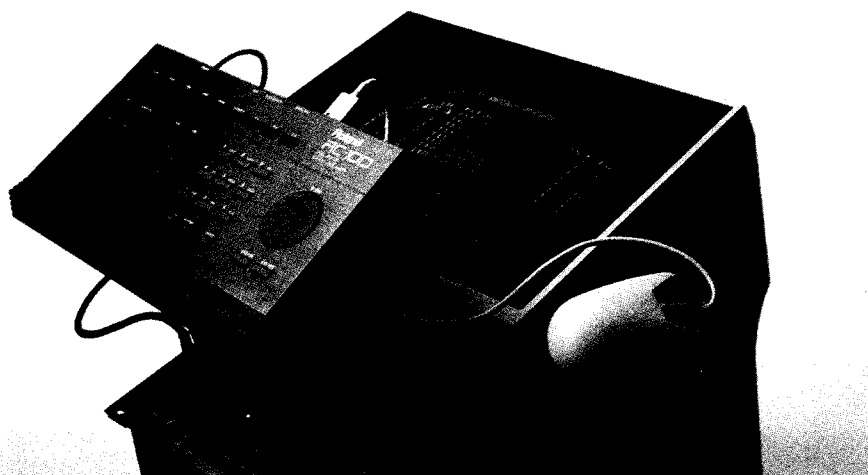



Foto 1

Foto 1: L'RC 100 della Roland è un tipico controllo a distanza per permettere il controllo globale delle voci dello strumento (S330-S550), velocizzandone la gestione, senza agire direttamente sul pannello

Alcune videate del programma Editor/ Librarian della serie Caged Artist Editors della Dr.Ts.

L'unica cosa che li accomuna è che tutti questi programmi per interagire con lo strumento, fanno uso dei messaggi MIDI di Sistema Esclusivo, che possono essere costituiti da un numero arbitrario di dati, a discrezione del costruttore.
A sua volta, in pratica, quasi ogni costruttore ha stabilito le proprie modalità di dialogo per il Sistema Esclusivo; è quindi difficile pensare ad un programma multipurpose.
Un raro esempio di editor universale è contenuto nell'ottimo pacchetto Music-X, da noi precedentemente descritto su questa stessa rivista.
Non vi spaventate comunque, perché quanto detto sopra è a puro titolo informativo: non è infatti necessario, per utilizzare tali programmi addentrarsi nei meandri dei messaggi di Sistema Esclusivo.
Per nostra fortuna infatti, l'interfaccia video verso l'utente è caratterizzata da grafici e tabelle su cui cliccare col mouse.
Senza che ve ne rendiate conto quindi, ad ogni azione di editing viene preparato in RAM il set di dati che, inviato allo strumento consente di suonare verificandone il risultato.
Quando siamo soddisfatti, possiamo assegnare un nome al nuovo suono creato e salvarlo su disco.
A questo punto si potrebbe pensare che, con un tool di questo tipo, siano state sfondate le vecchie barriere che limitavano la nostra creatività.
Ciò è vero, ma a volte si crea il problema di non sapere da che parte iniziare a creare un suono.
Se vi capita di essere a corto di immaginazione, il problema viene risolto con una funzione di cui è dotata la maggior parte dei programmi Editor: la generazione casuale, RANDOMIZE, di una parte (da definire) dei parametri.
In questo modo, senza troppa fatica, potrete fare lavorare il computer finché non vi proporrà qualcosa che ritenete interessante, o come prodotto finito oppure sul quale iniziare voi a lavorare di fino.
Creare voci per gli strumenti MIDI attraverso un computer è sicuramente interessante; se addirittura trovate il canale giusto e avete le doti di creatività necessarie, può costituire anche un'attività redditizia.
La capacità di creare nuovi effetti sonori da un moderno sintetizzatore digitale dipende oggi da alcuni fattori determinanti:

1- materiale base per la sintesi sonora
2- tipo e quantità di parametri di controllo dei suoni
3- gestibilità dei parametri
4- fantasia dell'autore

Il punto 1 dipende dal tipo di strumento scelto (sintesi FM, Analogica, campionamento, L.A., ecc.)
Il punto 2 dipende ancora dalla scelta dello strumento (per esempio: numero degli algoritmi, EG a 4,

Tabella 1

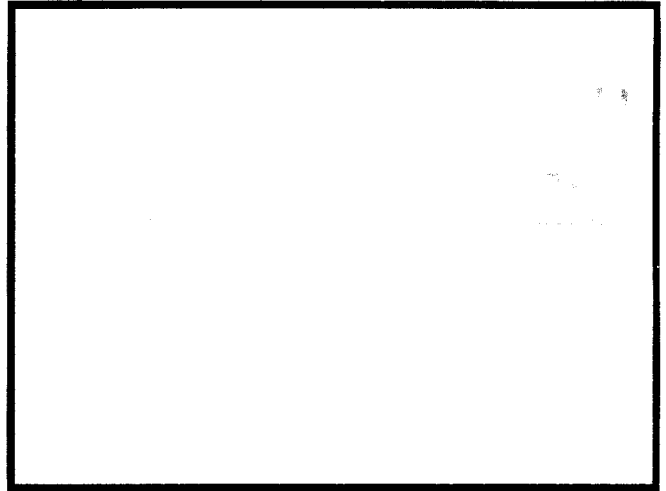
Tabella 1: Tipica tabella contenente tutti i parametri relativi ad una voce per il sintetizzatore Roland D-5C

Editor di voci Yamaha per il Sintetizzatore DX7. Sono evidenti i diagrammi che rappresentano graficamente i parametri da modificare

5 o più livelli, ecc.). E' riconosciuto da tutti che, se sui synth è difficile gestire i vari parametri, sugli expander è praticamente impossibile; esistendo, quindi, i software editor (e ce ne sono davvero molti!), adottatene subito uno.
Infine, quando tutto è pronto, siete nelle condizioni di dare libero sfogo alla vostra creatività.
Abbiamo accennato prima agli innumerevoli programmi Editor/ Librarian esistenti oggi sul mercato per Amiga. Una fra le ditte che, finora, ha sviluppato il maggior numero di programmi di questo tipo è indubbiamente la Dr. T's, per la serie Caged Artist's. A tutt'oggi, ci risultano disponibili, in questa collana, le seguenti versioni per i principali strumenti in commercio:

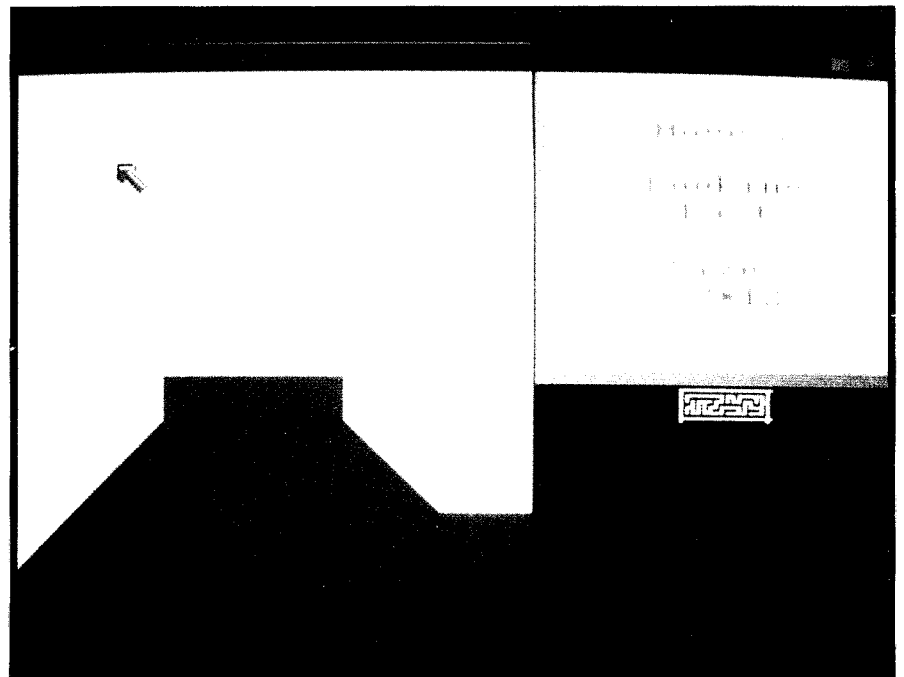
4-Op Deluxe (TX-81Z, FB-01, DX100, 27, 21)
Serie Casio CZ (CZ1, 101, 1000-5000)
Serie DX
Serie Ensoniq (ESQ-1/SQ80)
Kaway (K-1, K-3, K-5)
Lexicon PCM-70
Oberheim (MATRIX-6/1000)
Korg M-1
Roland (D-50, D-110, MT-32)

La serie Caged Artist's raccoglie, insomma, i migliori pacchetti Editor/ Librarian, per i migliori strumenti.
L'approccio di questi programmi è simile in ciascuno di essi: la fase di edit è caratterizzata da una videata colma di tabelle e di grafici, specifici per ogni synth, con i quali si può interagire in tempo reale con il mouse.
Una caratteristica molto interessante, che fa di questi software dei very "musician friendly" è la possibilità, sempre attraverso la pressione di un tasto sul mouse, di ascoltare una qualsiasi delle 88 note della scala, con il timbro quale è dopo l'ultima effettiva modifica, dando così al musicista la reale sensazione di ciò che sta creando.
La frequenza (pitch) e l'intensità (velocity) della nota, dipendono dalla posizione orizzontale e verticale del mouse. L'utilissima funzione Compare/Copy, presente, come tutte le altre, in ogni versione del programma, permette di fare un confronto diretto tra il suono (patch) editato e quello originale di partenza, oppure con ogni altro suono in memoria. Vengono evidenziati i parametri diversi fra i due suoni e ognuno di essi può essere copiato nell'altro.
Altra utile funzione è quella denominata Randomize, che permette, una volta selezionato un gruppo di parametri di ottenerne una variazione casuale; tale funzione, che abbiamo visto più sopra, aiuta notevolmente il musicista poco fantasioso; da notare che, in ogni momento, è possibile intervenire nella scelta della percentuale di casualità.
Tutti i software contengono le funzioni MIDI merge, Solo e ricanalizzazione (rechannelize), per semplificare l'uso di una master keyboard esterna o di un sequencer collegato. La maggior parte degli Editor Caged Artist's, è in grado di lavorare in Multitasking, offrendo così agli Editor la capacità di essere associati ad un sequencer.
Altro programma degno di nota, a questo proposito, non solo come editor, ma come primo esempio di Worstation MIDI è il già citato Music-X che, senza dubbio, rappresenta una svolta nel mondo del software MIDI per Amiga.
Fatta questa panoramica sul mondo dei programmi Editor/ Librarian, di cui abbiamo esaminato la caratteristiche e le potenzialità, vi lasciamo, dandovi appuntamento alla prossima puntata, in cui analizzaremo a fondo uno di questi programmi, l'Editor MT-32 della Caged Artist's, che abbiamo provato in accoppiata con il modulo MT-32 Roland ottenendo risultati davvero entusiasmanti.
Arrivederci alla prossima puntata.▲

ON DISK è una rubrica mensile di quattro pagine che possono anche essere staccate e conservate, in queste pagine sono descritte tutte le informazioni dei programmi inclusi nel disco, complete di istruzioni, trucchi ecc... In questo spazio troveranno posto giochi, utility e tutto ciò che può fare Amiga.

• Games

Maze

Maze è una piccola ma utile routine per creare labirinti. Il labirinto risultante ha la caratteristica di permettere uno e un solo percorso tra due dei suoi punti. La logica di funzionamento è abbastanza semplice: prima di tutto viene tracciato un finto passaggio rettangolare che delimita l'area in cui il labirinto deve svilupparsi, poi viene scelto un punto di partenza casuale. Il labirinto è diviso in quadratini. Dalla posizione di partenza viene scelta una direzione a caso. Se la direzione scelta conduce in un vicolo cieco è necessario prendere altre direzioni. Se, costruendo il labirinto, si ritorna al punto di partenza significa che il labirinto è terminato. I programmatori interessati ad approfondire lo studio di questa routine possono leggere il file-documento presente nella directory games. I giochi che utilizzano questa routine sono due: SimpleDemo e 3dDemo.

SimpleDemo si occupa di tracciare, su di uno schermo a bassa risoluzione, la mappa di un labirinto di 320 per 200 quadratini. I possessori di una stampante potranno utilizzare uno dei tanti programmi di screen-grabbing in circolazione per catturare l'immagine della mappa e portarla su carta (nel disco di Amiga Magazine 11 è stato pubblicato Snap!Plus, uno screen-grabber completo e pratico).Una volta stampata la mappa, potrete fissare a vostro piacere un punto di partenza e di arrivo (magari ai due angoli opposti della mappa), con la certezza che fra la partenza e l'arrivo esista uno e un solo percorso. Se non avete una stampante, non preoccupatevi, perché il secondo programma che utilizza MAZE è molto più completo e divertente. 3dDemo consente di entrare e muoversi all'interno di un labirinto tridimensionale. La grandezza del labirinto può essere decisa da menu e una volta dato il via al gioco è possibile muoversi utilizzando i tasti cursore. Un'altra opzione, sempre da menu, permette di esaminare la mappa del labirinto durante il gioco (la vostra posizione è indicata da un pixel evidenziato). Chi riuscirà ad uscire potrà sapere quanto tempo e quante mosse ha utilizzato per terminare la partita. E' stata implementata anche un'opzione di demo-play, che permette di osservare il computer mentre cerca la via d'uscita.

Boomerang

Il Boomerang è stato inventato molto tempo fa da tribù indigene dell'Australia per procurarsi del cibo, ma se utilizzato con precisione può risultare letale anche per l'uomo. E' da qui che prende ispirazione questo gioco. A Boomerang si gioca in due e necessita di due joystick.
La partita si svolge in un'arena che, solitamente, presenta dei muri posti casualmente in modo da ostacolare le possibili traiettorie delle armi dei due giocatori.
Una volta caricato il gioco, è comunque possibile scegliere di giocare in un'arena priva di muri premendo RETURN. Se invece si preme la barra spazio, verranno creati dei muri all'interno dell'arena, e una nuova configurazione di muri viene creata ad ogni pressione della barra; una volta trovata una configurazione di muri soddisfacente, premendo RETURN si può dare il via alla partita. Il giocatore verde inizia la partita nell'angolo in basso a sinistra, mentre quello viola parte dall'angolo in alto a destra.
Per muoversi è sufficiente utilizzare la leva del joystick e per lanciare il boomerang basta premere il tasto fire. Quando il boomerang è in volo, è possibile deviarlo utilizzando sempre la leva del joystick. Se il boomerang urta un muro, cade a terra ed è necessario andare a raccoglierlo, ma, con un po' di pratica, è possibile riprenderlo al volo. L'avversario, colpito dal boomerang, perde una vita e la partita riprende subito dopo che gli omini sono stati riportati alle loro posizioni di partenza. Ogni giocatore all'inizio della partita ha cinque vite; perde, naturalmente, chi finisce per primo le sue cinque vite.

• Utility

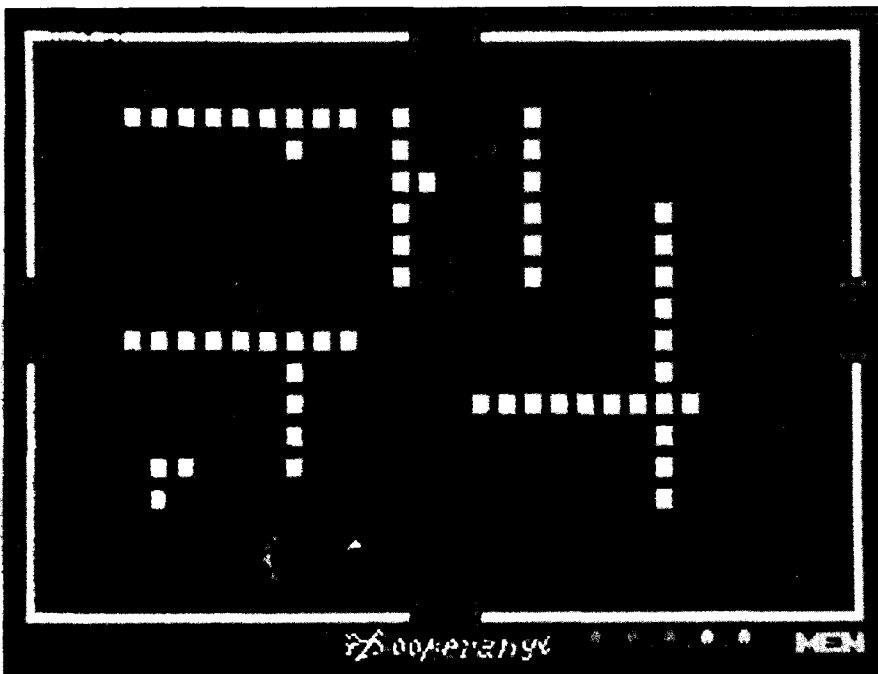
Turbo

Turbo è un piccolo programma che può venire in aiuto in tutti quei casi in cui il computer deve effettuare una moltitudine di calcoli che richiedono, normalmente, tempi molto lunghi.
Prendete, ad esempio, un programma per la generazione di immagini dell'insieme di Mandelbrot o di immagini tridimensionali con Ray-Tracing; in questo ultimo caso basta aggiungere qualche oggetto per moltiplicare enormemente i tempi di attesa che, a volte, possono protrarsi delle ore.
Il trucco utilizzato da Turbo per venirci in aiuto in queste situazioni consiste nel disattivare i bitplane, gli sprite, il copper e il DMA audio, in modo da lasciare all'elaborazione dati la quasi totalità del tempo macchina.
Una volta fatto girare il programma, appare una piccola finestra sullo schermo.
Cliccando sul gadget "TURBO!", l'utility disattiva lo schermo e tutto il resto; cliccando con il tasto sinistro all'esterno della finestra o premendo il tasto destro del mouse, tutto viene riattivato e si ritorna alle normali condizioni di lavoro.

StealMemBoot

Questa utility consente di modificare il Bootblock di un qualsiasi dischetto in modo che, dopo aver effettuato un booting con quel dischetto, venga utilizzata solo la cosiddetta Chip-RAM, cioè i primi 512 KByte di RAM. Questa caratteristica si rende utile per far funzionare correttamente tutti quei programmi che danno problemi se fatti girare in presenza di un'espansione di memoria.
Facendo girare il programma, StealMemBoot vi chiederà di inserire il disco, il cui BootBlock va modificato, nel drive DF0:.
Sarà quindi sufficiente rimuovere la protezione dalla scrittura, inserire il dischetto e premere RETURN.
Terminate le operazioni potrete poi provare a effettuare un Boot con il disco modificato: vi accorgerete che un messaggio indicherà che il BootBlock è stato modificato.

Attenzione!!

E' molto importante controllare il

MIDI: il software

Primi passi nella programmazione MIDI

Daniel Brookshier

In un numero precedente di Transactor per Amiga, Steve Simpson ha descritto i fondamenti dell'hardware MIDI e ha dato una breve occhiata al software elementare. In questo articolo, Dan va oltre, descrivendo il set di comandi MIDI e presentando alcune tecniche di programmazione MIDI (in Modula 2, ma adattabili ad altri linguaggi).

Perché si dovrebbe voler scrivere il proprio software MIDI? Nel mio caso, ho dei problemi nel canticchiare una canzone, per non parlare dei tentativi di suonare Beethoven o i Pink Floyd sulla mia tastiera Casio. Il mio interesse si colloca di più nell'area del software tipo Instant Music. Sogno un software che impedisca al novizio di eseguire musica che suoni come la tortura di orrendi animali.

Ho trovato molto poco tempo per dedicarmi a questo nobile compito. Così vi introdurrò nei segreti che ho imparato sul MIDI e Amiga, nella speranza che siate voi a salvare il mondo dalle note stonate.
Il programma di esempio è stato scritto in Benchmark Modula 2, ma è abbastanza semplice perché lo si possa convertire in C senza troppo lavoro. In ogni caso ci sono sufficienti informazioni nel testo, per scrivere software MIDI in linguaggi diversi dal C o dal Modula 2.

La parola di moda è MIDI

Il MIDI è apparso solo nei primi anni Ottanta, ma ora molte tastiere che costano meno di un centinaio di dollari circa, presentano sul retro delle connessioni MIDI.
Dunque, che cosa è esattamente il MIDI? Per iniziare, MIDI sta per Musical Instrument Digital Interface (interfaccia digitale per strumento musicale). Il MIDI permette a computer e strumenti musicali di interfacciarsi e controllarsi reciprocamente, usando un linguaggio di comunicazione standard. A causa della complessità dei sintetizzatori odierni e della necessità di collegarli ai computer o a speciali dispositivi, chiamati sequencer, il MIDI è divenuto una necessità.

L'uso del MIDI può anche aiutare indirettamente nella composizione musicale, permettendo al compositore di registrare la musica direttamente dalla tastiera del sintetizzatore. Una volta che un brano musicale sia stato registrato, può essere riprodotto o modificato. Durante un concerto dal vivo, si possono modificare premendo un pulsante gruppi interi di caratteristiche di uno strumento, riducendo gli oneri del musicista, che potrebbe avere bisogno di modificare dozzine di parametri di uno strumento mentre sta suonando.

Il MIDI e l'interfaccia seriale

Su uno strumento MIDI di base ci sono tre connettori. Questi sono prese DIN a cinque pin che sono etichettate IN, OUT e THRU.
La presa IN, come implica il suo nome, riceve dati e dovrebbe normalmente essere connessa all'OUT di un altro strumento (o computer) che stia inviando messaggi allo strumento.
Il connettore OUT è usato quando si devono mandare messaggi da questo strumento verso altri strumenti o computer.
Il connettore THRU serve solamente a far passare oltre i segnali che giungono dal connetore IN; permette ad una sorgente il controllo di più strumenti.

Leggere e scrivere attraverso un adattatore MIDI posto sulla linea seriale di Amiga è facile. Il sorgente midi.mod è un modulo, scritto in Modula 2, che contiene tutte le procedure necessarie ad aprire, chiudere, leggere e scrivere mediante il MIDI.
L'uso del MIDI è molto simile a quello di una normale comunicazione seriale, come potrebbe avvenire via modem. Infatti i parametri per il MIDI sono gli stessi della normale trasmissione mediante RS232: otto bit di dati con un bit di stop e nessuna parità. Le uniche differenze stanno nell'ioSerFlags del device seriale e nella speciale baud rate. Sia il flag SerRadBoogie che SerXDisabled devono essere impostati e la baud rate deve essere pari a 31.25 Kilobaud. Con una baud rate così alta il device seriale deve funzionare nella maniera più efficiente possibile e l'impostazione del flag SerRadBoogie costringe il device seriale a saltare molto codice interno di controllo. Dopo aver inizializzato il device seriale, la lettura o la scrittura dei codici MIDI sono realizzate inviando dei messaggi al Port del device.

La composizione di un messaggio MIDI

I segnali che gli strumenti MIDI inviano o ricevono hanno cinque formati. Tutti consistono di almeno un byte, sebbene la maggior parte abbia una lunghezza di due o tre byte. Il primo byte è il byte di stato che viene mandato per segnalare un nuovo messaggio o un nuovo evento. Il suo primo bit (il più significativo) è impostato. Lo si può controllare verificando se il valore del byte è maggiore di 127.

Il byte di stato può essere seguito da uno o più byte di dati che hanno il bit più significativo azzerato. Questo significa che noi possiamo stabilire se si tratta di un byte di dati, semplicemente controllando se questo bit è a zero o verificando che il valore dell'intero byte è minore o uguale a 127.
Un modo per determinare che cosa si deve fare quando si riceve un byte dal MIDI è usare la seguente sequenza:

```
IF MIDI_In > 127 THEN
  (* trattamento del byte di stato *)
ELSE
  (* elaborazione del byte di dati *)
END;
```

Ora dobbiamo decodificare il byte di stato. Ci sono attualmente cinque tipi di byte di stato, ognuno dei quali riguarda un differente livello di controlli. La figura 1 fornisce i modelli dei bit di questi messaggi, assieme al formato dei dati. Fra questi ci sono due gruppi fondamentali di messaggi: messaggi di canale (channel message) e i messaggi di sistema (system message). Per iniziare la decodifica, si deve prima determinare il gruppo specifico cui appartiene il messaggio.

Il byte di stato viene prima diviso in due sezioni: una metà alta e una bassa. Il tipo e il gruppo del comando stanno nella metà alta, la più significativa. La metà bassa contiene o il numero di canale o il sottotipo del comando. Questi numeri possono essere decodificati in molti modi. Il primo modo è quello mettere in due variabili distinte, mediante una maschera, il tipo e il numero di canale. In C può essere molto semplice. Basta in pratica fare un AND fra una costante e il byte di stato:

```
Command = 0xF0 & MIDI_In;
Channel = 0x0F & MIDI_In;
```

In Modula 2 è necessario un approccio diverso. Per fare un AND in Modula 2 dobbiamo prima trasformare il byte di stato in un set. Dopo di che vengono determinate le intersezioni tra la maschere e i set di stato e il risultato viene convertito di nuovo in un byte senza segno.
Nell'esempio che segue, i valori che devono subire l'AND devono essere convertiti in set perché il compilatore possa interpretare il segno "*" come un'istruzione AND. Poi il valore è convertito in tipo CARDINAL (intero senza segno). L'operazione è la stessa che in C, solo la terminologia è cambiata. Inoltre, la metà più alta viene spostata sulla destra in modo che il numero sia compreso nella gamma 8-15. Ciò rende il codice più semplice a leggersi in momenti successivi.

```
TYPE
  MaskSet : SET OF [0..7];
VAR
  MIDI_In : BYTE;
  Command, Channel : CARDINAL
BEGIN
  CommandMSB = CARDINAL(SHIFT(MIDI_In,-4));
  CommandLSB =
     CARDINAL(MaskSet(00FH) * MaskSet(MIDI_In));
```

Ora che il comando è stato diviso in blocchi da 4 bit, è semplice determinare il gruppo e il comando del messaggio. La procedura DecodeMIDICommand svolge la maggior parte del lavoro: estrae i dati dal byte di stato, poi usa istruzioni case per determinare il tipo del comando. Come si può vedere dalla figura 1, nella metà più alta del byte di stato i valori dei messaggi di canale variano tra 8 e 14 ($E), mentre i messaggi di sistema hanno tutti il valore 15 ($F). Con i messaggi di canale la metà bassa del byte di stato contiene il numero di canale, mentre per i messaggi di sistema, la metà più bassa del byte di stato contiene l'effettivo comando di sistema.

Ci sono quattro tipi diversi di azione da effettuare, a seconda del tipo e del gruppo del comando. In primo luogo, il comando può non essere implementato, nel qual caso si chiama una procedura per avvisare l'utente che è avvenuto un errore.
La seconda possibilità è che il comando sia seguito da uno o più byte di dati. In questo caso, si stabilisce il numero di byte che ci si deve aspettare e si azzera il contatore dei byte di dati. Il numero di canale e quello del comando vengono salvati, per essere usati quando tutti i dati siano stati ricevuti.

La terza possibilità è che sia stato ricevuto un comando che non necessita di altri dati. In questo caso, siccome non dobbiamo aspettare altri dati dall'interfaccia MIDI, si può effettuare immediatamente l'azione prevista dal comando, chiamando un'altra procedura. Ciò può causare la quarta azione possibile.
Se il programma non implementa la funzione, allora si può chiamare una procedura alternativa per informare l'utente che la funzione non è supportata. Tuttavia, si deve fare attenzione a stampare questo tipo di messaggio, in quanto alcuni dei comandi possono giungere tanto velocemente quanto è possibile leggerli; pertanto la stampa di messaggi relativi alle funzioni non implementate dovrebbe essere fatta solo in modo debug!

Sia per i comandi errati, che per quelli non implementati, viene impostata una variabile per consentire al programma di ignorare ogni dato che può essere stato inviato dopo di essi.
Ora che il byte di stato è stato decodificato, si devono

Comando	Byte di stato		Primo dato	Sec. dato
Messaggi di voce di canale				
note off	$8n	1000nnnn	0yyyyyyy numero della nota	0yyyyyyy velocità di rilascio
note on	$9n	1001nnnn	0yyyyyyy numero della nota	0yyyyyyy velocità di attacco
polyphonic key pressure	$An	1010nnnn	0yyyyyyy numero della nota	0yyyyyyy aftertouch
control change (*)	$Bn	1011nnnn	0yyyyyyy n. del controllo	0yyyyyyy pos. del controllo
program change	$Cn	1100nnnn	0yyyyyyy n. di programma	
channel pressure	$Dn	1101nnnn	0yyyyyyy aftertouch	
pitch bend	$En	1110nnnn	0yyyyyyy pos. del bender (byte basso)	0yyyyyyy pos. del bender (byte alto)
Messaggi di modo di canale				
local control	$Bn	1011nnnn	01111010	01111111=on 00000000=off
all note off	$Bn	1011nnnn	01111011	00000000
omni off	$Bn	1011nnnn	01111100	00000000
omni on	$Bn	1011nnnn	01111101	00000000
mono on (poly off)	$Bn	1011nnnn	01111110	0000vvvv
poly on (mono off)	$Bn	1011nnnn	01111111	00000000
Messaggi di sistema comuni				
song position pointer	$F2	11110010	0yyyyyyy pos. della song (byte basso)	0yyyyyyy pos. della song (byte alto)
song select	$F3	11110011	0yyyyyyy numero della song	
tune request	$F6	11110110		
Messaggi di sistema in tempo reale				
MIDI clock	$F8	11111000		
start	$FA	11111010		
continue	$FB	11111011		
stop	$FC	11111100		
active sensing	$FE	11111110		
system RESET	$FF	11111111		
Messaggi di sistema esclusivi				
system exclusive	$F0	11110000	0yyyyyyy identif. produttore	0yyyyyyy dati non standard
end system exclusive	$F7	11110111		

Note:

n e nnnn	indicano il numero di canale MIDI
yyyyyyy	indica dati arbitrari
vvvv	indica il numero di canale a cui assegnare la voce
(*)	si veda la figura 2

Figura 1: Sommario dei comandi MIDI

rispettare le seguenti regole del protocollo:

Regola 1. Ciascun byte di stato, se necessita di dati, attende un numero specifico di dati. Nessun'azione dovrebbe essere eseguita, finché non siano stati ricevuti tutti i dati.

Regola 2. Dopo che un byte di stato è stato ricevuto, può essere inviato un numero qualsiasi di messaggi di dati completi, senza ricevere un altro byte di stato.

Regola 3. I messaggi di sistema possono essere ricevuti in qualsiasi momento, anche tra byte di dati!

Fra queste, la regola 2 è la più importante. Il modo migliore per spiegarla è mediante un esempio. Se un tasto è premuto su una tastiera di un sintetizzatore, la tastiera invierà un byte di stato e due byte di dati. Se vengono premuti due tasti nello stesso momento, allora il dispositivo MIDI può inviare un byte di stato e due gruppi di due byte.

La regola 2 genera problemi con una variante del comando Note On. Su alcuni strumenti, il comando Note Off non viene usato; invece, un comando Note On con velocità zero significa che la nota si è conclusa. Questa è un'esigenza che un programma MIDI dovrebbe prendere in considerazione, per evitare di lasciare delle note nella posizione on.

Questo significa che, se viene premuto un tasto su questo tipo di strumento prima che il vostro programma sia iniziato, il programma non avrà potuto vedere il byte di stato e continuerà a ricevere dei dati di cui non conosce il significato. Un programma MIDI dovrebbe sempre ignorare i dati fino a quando non riceva un byte di stato. Il solo problema è che potreste aver bisogno di premere un tasto funzione dello strumento, affinché venga inviato un byte di stato la prossima volta che sarà premuto un tasto della tastiera.

Il modulo principale ReadallMIDI legge i comandi fondamentali note e pitch bend ed emette un'onda triangolare su uno o più dei quattro canali del device audio. Quando viene ricevuto un comando, il valore della nota viene convertito in una frequenza e viene determinata l'ottava. Se un canale è aperto, si fa iniziare il suono su quel determinato canale; se si muove il comando di controllo della tonalità, il nuovo valore viene aggiunto alla lunghezza d'onda da suonare. Quando il tasto dello strumento MIDI viene rilasciato, il suono del canale corrispondente viene fermato. Se dovete aggiungere al codice nuovi comandi di canale MIDI, basta un'istruzione CASE e il valore numerico del comando.

I comandi MIDI

Prima di descrivere il set di comandi MIDI (se ne vede un sommario in figura 1), dovrei avvisare chi è in procinto di comprare uno strumento controllabile via MIDI, di non comprare mai uno strumento MIDI, senza verificare, sul manuale, le funzioni che supporta. Che la tastiera sia in grado di fare certe cose non significa che esse possano

essere pilotate completamente attraverso il MIDI. Sul Casio HT-700, per esempio, il novanta per cento delle funzioni non sono sfruttabili attraverso l'interfaccia MIDI. Il problema principale è che è possibile personalizzare i suoni dello strumento solo attraverso dispositivi a rotazione. Per cambiare le caratteristiche di un solo strumento ci vorrebbero molti minuti.

Ciò ci conduce al primo e più importante comando per coloro che vogliano esaminare strumenti di alto livello: il messaggio di sistema esclusivo (system esclusive). Come tutti gli altri messaggi, questo inizia con un byte di stato; poi arriva un byte di dati che contiene l'identificazione della casa produttrice. A questo fa seguito una serie di byte, che contiene dati speciali sullo specifico strumento e il comando viene completato da un byte di stato che indica la fine del sistema esclusivo (end system exclusive). Sebbene esistano alcuni formati standard, comuni alle compagnie, dovreste riferirvi al vostro manuale per avere ulteriori informazioni.

I comandi più comuni sono i messaggi di canale. Questi usano il numero di identificazione del canale che viene ricavato dalla metà più bassa del byte di stato. Il numero varia tra 0 e 15; il che significa che fino a 16 strumenti MIDI possono essere controllati da un cavo MIDI. Il numero di canale rivela quale strumento, o quale parte dello strumento, sta inviando o ricevendo i dati. Il numero del canale deve essere elaborato, perché è possibile ricevere un messaggio di canale da uno strumento che vogliamo ignorare. Si ricordi che il MIDI permette la connessione di molti strumenti con lo stesso cavo! Sulla tastiera del Casio HT-700 anche i numeri di canale della tastiera ritmica e di quella principale sono differenti.

I principali messaggi di canale riguardano la nota che deve essere suonata e come deve essere controllata. Note on e note off specificano la nota da suonare o da fermare. I byte di dati di entrambi i comandi includono sia la nota che la velocità del tocco. La nota si determina partendo dal valore 60 che corrisponde al DO fondamentale; l'addizione o la sottrazione di una unità a questo numero corrisponde all'addizione o alla sottrazione di un semitono. Su un piano, un semitono è la distanza tra tasti bianchi e neri adiacenti, tra il tasto del SI e quello del DO e tra quello del MI e quello del FA.

Il valore del tasto MIDI deve essere convertito in una frequenza per essere usato dal device audio di Amiga. La procedura MIDI2FreqConv viene usata per formare una tavola di frequenze per ciascuna nota. L'equazione usata è molto semplice, si moltiplica l'ultimo valore della tavola per una costante, ma i risultati sono molto accurati. Il numero del tasto MIDI è usato come un indice per la tavola. La velocità della nota è la forza con cui la nota viene suonata. Un esempio di ciò potrebbe essere la forza usata quando si preme un tasto del piano. Se il tasto è premuto con un tocco

Controlli continui (byte più significativo)

0	indefinito
1	Modulation wheel
2	Breath controller
4	Foot controller
5	Portamento time
6	Dato (byte più significativo)
7	Volume principale
8	Bilanciamento
10	Pan
11	Expression controller
12-31	controlli generali

Controlli continui (byte meno significativo)

32	indefinito
33	Modulation wheel
34	Breath controller
36	Foot controller
37	Portamento time
38	Dato (byte meno significativo)
39	Volume principale
40	Bilanciamento
42	Pan
43	Expression controller
44-63	controlli generali

Note:

* Se un controllo continuo richiede un dato non compreso nella gamma 0-127, possono essere inviati due parametri di controllo, usando differenti numeri per lo stesso dispositivo; uno conterrà il byte più significativo del dato e l'altro il meno significativo.
* Per i dati nella gamma 0-127, sono usati solo i numeri del dispositivo relativi al byte più significativo.
* Alcune case produttrici usano controlli non standard e perciò anche numeri di controllo non standard.

Controlli a interruttore (Dati: on=127, off=0)

64	Sustain pedal
65	Portamento
66	Sostenuto
67	Soft pedal
69	Hold 2
80-95	controlli generali

Controlli diversi

96	Incremento dei dati
97	Decremento dei dati
98	Parametro non registrato (bit bassi)
99	Parametro non registrato (bit alti)
100	Parametro registrato (bit bassi)
101	Parametro registrato (bit alti)
102-121	indefiniti
122-127	riservati ai messaggi di modo (si veda la figura 1)

Figura 2: Valori dei parametri di controllo

leggero, il suono prodotto ha un volume basso. Se il tasto è premuto con forza, il suono prodotto ha un volume più alto. I dispositivi elettronici aggiuntivi, che occorrono affinché una tastiera possa registrare la velocità del tocco, sono così costosi che alcune tastiere contengono solo interruttori on/off.

Il comando note off è simile a note on. La sola differenza è che la velocità indica, ora, con quale velocità il suono diminuisce fino al silenzio. Si ricordi che questo comando può essere sostituito da una velocità zero in un comando note on.

Il comando polyphonic key pressure è un altro di quei comandi che si trovano solo sulle tastiere più care. Esso indica quanta pressione costante viene esercitata quando si suona una determinata nota. Nella maggior parte delle spiegazioni relative a questo comando si afferma che esso viene usato per indicare il vibrato. Può anche essere usata negli strumenti a fiato più recenti, come il nuovo sassofono MIDI. In questo caso il valore della pressione vorrebbe indicare con che forza si sta soffiando nello strumento.

Il comando program change è spesso chiamato il numero di patch. Usando questo comando negli strumenti che possiedono più strumenti preprogrammati, si può cambiare lo strumento selezionato. Su alcune tastiere, una parte dei valori è anche usata per selezionare il ritmo.

Il comando control change è piuttosto versatile e permette il controllo di molti parametri diversi. La figura 2 elenca alcuni dei possibili valori che possono essere modificati. La selezione del parametro si trova nel primo byte di dati e il nuovo valore per il parametro si trova nel secondo byte. Si ricordi che il vostro strumento potrebbe non riconoscerli tutti.

Il comando channel pressure è simile al comando polyphonic key pressure, tranne per il fatto che, mentre polyphonic key pressure rappresenta la pressione applicata su una singola nota, channel pressure influenza tutte le note che vengono correntemente suonate.

L'ultimo messaggio di canale è il pitch bend. Questo è uno dei comandi più divertenti e si trova anche sulla maggior parte delle tastiere di recente produzione. Il codice rappresenta la posizione del pitch bend, un dispositivo di controllo che viene spesso usato per modificare la tonalità di una nota (verso l'alto o verso il basso) mentre viene suonata.

Il dato del pitch bend comincia con il valore 8192; ciò significa che viene sottratto 8192 al valore del pitch bend e il risultato viene aggiunto a tutte le note in corso di esecuzione. Per l'HT-700 e molte altre tastiere, viene usato solo il byte più significativo e così i dati iniziano con il valore 64. Nel programma di esempio il valore del pitch bend viene aggiunto direttamente al periodo inviato al device audio.

I messaggi di modo di canale (channel mode) sono usati per controllare le caratteristiche della tastiera. Il comando local control off permette di usare la tastiera come un dispositivo di input, senza suonare nessuno dei suoi strumenti. Questo è utile quando si usa la tastiera solo per mandare dati ad Amiga o ad altri dispositivi MIDI.

Il prossimo messaggio di modo di canale è il comando all notes off. Questo fa esattamente ciò che significa: ferma l'esecuzione di tutte le note. E' utile quando qualche nota rimane in esecuzione.

Il resto dei messaggi di modo di canale riguardano il numero di note che possono essere suonate contemporaneamente. I comandi poly permettono a due o più note di essere suonate allo stesso tempo, mentre il modo omni permette di suonare una sola nota alla volta. Se vengono inviate più note di quante se ne attendono, le ultime note verranno ignorate.

Messaggi di sistema in tempo reale

Il prossimo gruppo di comandi MIDI è costituito dai messaggi di sistema in tempo reale (system real time). Essi consistono di un solo byte e possono apparire in mezzo ad altri messaggi. E' molto importante ricordarlo. Se state ricevendo eventi relativi alle note e poi ricevete un messaggio di sistema in tempo reale, il prossimo byte di dati dovrebbe essere ancora trattato come un evento relativo alle note. Tutti questi messaggi sono associati con i sequencer e controllano la partenza, la conclusione e la sincronizzazione di eventi.

Il più importante di questi comandi è il segnale di sincronismo che viene inviato ogni volta che la luce lampeggia sulla sezione ritmica della maggior parte delle tastiere. Il segnale è usato come metronomo. E' usato il più delle volte come aggiornamento costante del sincronismo. Siccome non tutti gli strumenti sono identici, è necessaria una sorgente comune di sincronizzazione. Uno strumento con una sezione ritmica dovrebbe avere un interruttore per passare dalla ricezione all'invio di questo comando.

Il MIDI è un'interfaccia potente che consente molta flessibilità. L'esempio di questo articolo non raggiunge l'obiettivo di usare il set completo dei comandi possibili, ma questi possono essere aggiunti con pochissimo lavoro. Basta ricordare le regole e non suonare delle note stonate.

Gli interrupt di Amiga

Una guida agli interrupt e al loro uso

Ian Potts

Ian Potts è un programmatore freelance che lavora per una casa di produzione di software. Programma da 7 anni ed è l'autore di molti giochi commerciali per il Commodore 64 e Amiga. Ian sta lavorando attualmente ad un nuovo prodotto di intrattenimento per Amiga e ad un word processor. E' uno dei pochi programmatori di giochi per Amiga che usa il sistema operativo legalmente, invece di gestire direttamente l'hardware.

Il supporto hardware per gli interrupt

La CPU di Amiga (il 68000) ha molti livelli di interrupt, noti come eccezioni (exception). Questi vanno dal livello 1 fino al livello 7, quello del Non Maskable Interrupt (interrupt non mascherabile). Come per altre eccezioni del 68000, i vettori per gli interrupt sono posti nel primo Kbyte della memoria di Amiga, dalla locazione $64 a $7C.
Quando si verifica un interrupt, il computer spinge il registro di stato e il program counter sul stack utente corrente e salta all'appropriato codice di interrupt attraverso uno di questi vettori.

I sette livelli di interrupt hanno un ordine di priorità tale che un interrupt di livello più alto può interrompere un interrupt di livello più basso, ma un interrupt di livello più basso non può interrompere un interrupt di livello più alto.
In Amiga il chip Paula complica un po' le cose, in quanto fornisce attualmente quattordici livelli di interrupt invece di sette. Questi quattordici interrupt sono derivati dai livelli di interrupt della CPU, in modo che più sorgenti di interrupt possono condividere lo stesso interrupt della CPU come si vede nella figura 1.

Il Non Maskable Interrupt (NMI) può essere generato da hardware esterno e genera un interrupt di livello 7 il cui vettore si trova a $7C.
Quando si verifica un Vertical Blank (quando cioè il pennello elettronico raggiunge il punto inferiore dello schermo e ricomincia il suo percorso a partire dall'alto) viene generato un interrupt di livello 3: il processore interrompe quello che stava facendo, salva il registro di stato e il program counter sullo stack utente e salta alla routine il cui indirizzo si trova in $6C. Poi esegue questa routine che si conclude con un'istruzione RTE (ReTurn from Exception, ritorno da un'eccezione), che recupera il program counter e il registro di stato e permette al processore di continuare.
Siccome la routine di interrupt di livello 3 può essere causata da un interrupt Copper, Vertical Blank o Blitter Finished (conclusione del Blitter), allora la routine deve essere in grado di controllare quale interrupt sia avvenuto per agire in maniera appropriata. Ciò avviene leggendo i registri di interrupt di Paula. Essi sono:

```
INTREQ   $DFF09C INTerrupt REQuest (richiesta) (scrittura)
INTREQR $DFF01E INTerrupt REQuest (lettura)
INTENA   $DFF09A INTerrupt ENAble (abilitazione) (scrittura)
INTENAR $DFF01C INTerrupt ENAble (lettura)
```

L'impostare o l'azzerare i bit di INTENA abilita o disabilita la corrispondente sorgente di interrupt. I bit corrispondono a quelli mostrati in figura 1; cioè il bit 0 è TBE, il bit 3 è VERTB e così via.

Interrupt Paula — **CPU**

Bit	Nome	Significato	Livello	Vet.
00	TBE	Transmit Buffer Empty (buffer di trasf. vuoto)	1	$64
01	DSKBLK	DiSK BLocK (conclusione di blocco del disco)	1	$64
02	SOFT	SOFTware interrupt	1	$64
03	PORTS	PORTS (porte I/O e timer)	2	$68
04	COPER	COPPER	3o	$6co
			6	$78
05	VERTB	VERTical Blank	3	$6C
06	BLIT	BLITter finished (conclusione del blitter)	3	$6C
07	AUD0	canale AUDio 0	4	$70
08	AUD1	canale AUDio 1	4	$70
09	AUD2	canale AUDio 2	4	$70
10	AUD3	canale AUDio 3	4	$70
11	RBF	Receive Buffer Full (buffer di ricezione pieno)	5	$74
12	DSKSYN	DiSK SYNc (byte di sincronia disco trovato)	5	$74
13	EXTER	EXTERnal (interrupt esterni)	6	$78
14	INTEN	master INTerrupt ENAble (abilitazione interrup)	-	-

Figura 1: Gli interrupt di Amiga

Il bit 14 (INTEN) è il bit generale (master) di abilitazione. Se questo è pari a zero, le routine degli interrupt in ROM termineranno immediatamente senza eseguire il codice di interrupt.Per impostare o azzerare i bit di INTENA, si scriva un 1 nel bit o nei bit desiderati, con il bit 15 (bit di SET/CLR) pari a 1 per impostare i bit, o a 0 per azzerarli.

Il registro INTENAR vi permette di leggere lo stato corrente degli interrupt. I registri INTREQ e INTREQR vi permettono o di causare un interrupt, impostando dei bit in INTREQ, o di controllare quali interrupt si sono verificati, leggendo INTREQR.
In figura 2 appare la routine della ROM di Amiga che viene chiamata attraverso il vettore di livello 3 posto in $6c.

Dapprima, la routine salva molti registri sullo stack. Poi, controlla se il bit Master Interrupt Enabled (INTEN) (abilitazione principale degli interrupt) è impostato nel registro INTENAR di Paula ($DFF01C). Se è zero, ritorna, in quanto ciò significa che tutti gli interrupt sono stati disabilitati. Altrimenti effettua un AND fra il valore di INTENAR e il contenuto di INTREQR ($DFF01E), che contiene i bit degli interrupt che si sono verificati. Poi la routine continua, controllando quale bit sia impostato; se è impostato il bit 6 (Blitter Finished), chiama la routine per l'interrupt Blitter Finished, se è impostato il bit 5, chiama quella per l'interrupt Vertical Blank e se è impostato il bit 4, chiama quella per l'interrupt Copper. Se nessuno di questi interrupt si è verificato, la routine fa ritorno.

Ogni altro vettore per gli interrupt hardware chiama routine simili, che controllano i bit di Paula per decidere quale interrupt si è verificato e agire di conseguenza.

Le strutture Interrupt Vector

La figura 2 rivela che, dopo che la routine degli interrupt di livello 3, posta nella ROM di Amiga, ha deciso quale interrupt Paula si è verificato, vengono eseguite 3 importanti istruzioni. Nel caso dell'interrupt Vertical Blank queste sono:

```
MOVEM.L $90(A6),A1/A5  ;IVVERTB: IV_DATA in A1,
                        IV_CODE in A5
PEA     -$24(A6) ;indirizzo di ExitIntr() sullo stack
```

A questo punto, A6 contiene l'indirizzo della struttura ExecBase (descritta nel file include exec/execbase.i). La prima istruzione fa riferimento alla struttura Interrupt Vector, IVVERTB, contenuta in ExecBase. Questa struttura [nota in C come struttura IntVector, N.d.T] è definita nel file include exec/interrupts.i a questo modo:

```
STRUCTURE IV,0
 APTR  IV_DATA  ;indirizzo dei dati da passare alla
                 routine di interrupt
 APTR  IV_VODE ;indirizzo della routine di interrupt
               da chiamare
APTR   IV_NODE;indirizzo della struttura dell'inter
              rupt server (o nodo)
             ;usata per installare questo Interrupt
              Vector
quando è
             ;stata chiamata SetIntVector()
LABEL IV_SIZE ;lunghezza della struttura Interrupt
              Vector
```

Questa struttura è una struttura privata di Exec, usata per saltare alle routine di interrupt [le strutture private sono proprio tali e il loro uso è riservato al sistema; tutto ciò che

Parte del codice per gli interrupt di livello 3 di Amiga

```
; Routine per gli interrupt di livello 3, chiamata attraverso il vettore
; hardware posto a $6C
; Disassemblata dal KickStart 1.2 (c) Commodore Amiga Inc.

FC0CD8 MOVEM.L D0-1/A0-1/A5-6,-(A7) ; Salviamo i registri
       LEA      $DFF000,A0        ;custom in A0
       MOVEA.L  4,A6      ;ExecBase in A6
       MOVE.W   $1C(A0),D1 ;INTENAR
       BTST     #$0E,D1   ;Master Interrupt Enabled?
       BEQ      $FC0C4C   ;se no, esce.
       AND.W    $1E(A0),D1;AND con INTREQ per controllare se gli
                          ;interrupt occorsi erano abilitati
       BTST     #6,D1     ;E' l'interrupt Blitter Finished?
       BEQ.S    $FC0D08   ;se no, salta.
       MOVEM.L $9C(A6),A1/A5   ;IVBLIT: IV_DATA in A1,
                                IV_CODE in A5
       PEA      -$24(A6)  ;indirizzo di ExitIntr() sullo stack
       JMP      (A5)      ;chiama la routine dell'Interrupt Vector
                          ;relativa al Blitter
FC0D08 BTST     #5,D1     ;E' l'interrupt Vertical Blank?
       BEQ.S    $FC0D1A   ;se no, salta.
       MOVEM.L $90(A6),A1/A5   ;IVVERTB: IV_DATA in A1,
                                IV_CODE in A5
       PEA      -$24(A6)  ;indirizzo di ExitIntr() sullo stack
       JMP      (A5)      ;chiama la routine dell'Interrupt Vector
                          ;relativa al Vertical Blank
FC0D1A BTST     #4,D1     ;Interrupt di livello 3 del Copper?
       BEQ.S    $FC0D2C   ;se no, salta.
       MOVEM.L $84(A6),A1/A5   ;IVCOPER: IV_DATA in A1,
                                IV_CODE in A5
       PEA      -$24(A6)  ;indirizzo di ExitIntr() sullo stack
       JMP      (A5)      ;chiama la routine dell'Interrupt Vector
                          ;relativa al Copper
FC0D2C BRA      $FC0C4C   ;Interrupt sconosciuto, esce.
                          ; routine di uscita
FC0C4C MOVEM.L (A7)+,D0-1/A-1/A5-6 ;Recupera i registridallo
                                    stack
       RTE                ;ritorno dall'eccezione
```

Figura 2

Parte della routine di gestione delle catene di interrupt server di Amiga

```
;--- Routine Exec di gestione della catena di interrupt server, chiamata
;--- attraverso le strutture Interrupt Vector del Vertical Blank e del Copper
;--- Disassemblata dal KickStart 1.2 (c) Commodore Amiga Inc.
FC12FC: MOVE.W  $12(A1),-(A7)  ;Bit di INTREQ da azzerare in uscita
        MOVE.L  A2,-(A7)       ;Salva A2 sullo stack
        MOVEA.L (A1),A2        ;Indirizzo del primo server nella catena
                               ;(a più alta priorità)
FC1304: MOVE.L  (A2),D0        ;LN_SUCC - se non c'è successore, allora
        BEQ.S   $FC1316        ;è il server vuoto di fine lista, esce.
        MOVEM.L $0E(A2),A1/A5  ;IS_DATA in A1, IS_CODE in A5
        JSR     (a5)           ;chiama il codice del server
        BNE.S   $FC1316        ;D0 non è 0, così interrompe la catena
        MOVEA.L (A2),A2        ;LN_SUCC - ricava il prossimo server
                               ;della catena
        BRA.S   $FC1304        ;e ricomincia il ciclo
FC1316: MOVEA.L (A7)+,A2       ;Recupera A2 dallo stack
        MOVE.W  (A7)+,$DFF09C  ;Azzera i bit di INTREQ
        RTS                    ;Tutto fatto.
```

Figura 3

è privato può cambiare in versioni future del sistema operativo, come la 2.0 ex 1.4].

Nella prima istruzione citata, il campo IV_DATA della struttura è messo in A1 e il campo IV_CODE in A5. Il campo IV_CODE contiene l'indirizzo della routine a cui saltare quando si verifica un interrupt e il campo IV_DATA contiene l'indirizzo dei dati accessibili a quella routine. Nel caso del Vertical Blank, IV_CODE punta alla routine Exec di gestione della catena di interrupt server (si veda la figura 3) e IV_DATA punta a un nodo di intestazione di lista che sta all'inizio di una lista di routine server, le quali dovrebbero essere chiamate durante il Vertical Blank.

La seconda istruzione ripone sullo stack l'indirizzo del vettore di salto alla routine ExitIntr( ) di Exec. Questa routine sarà chiamata quando il codice dell'Interrupt Vector avrà terminato. E' una routine privata di Exec, chiamata dopo ogni interrupt, che sovraintende, essenzialmente, alla capacità multitasking di Amiga di passare il controllo ad un altro task dopo un determinato periodo di tempo.

La terza istruzione, infine, salta alla routine puntata da A5, che è l'indirizzo IV_CODE derivato dalla struttura Interrupt Vector. Nel caso del Vertical Blank (e di molti altri interrupt, come quello del Copper) si salta alla routine della figura 3. Come abbiamo visto, ci sono due tipi di vettori di interrupt: i vettori hardware ($64-$7c) e i vettori Exec contenuti in ExecBase. In generale, non dovreste mai alterare i vettori di interrupt hardware, perché ciò interromperebbe il multitasking. Per le applicazioni che si impadroniscono dell'intera macchina (come i giochi) questo potrebbere essere accettabile, ma solitamente non è necessario.Le strutture Interrupt Vector in ExecBase possono essere alterate, perché puntino alla vostra routine di interrupt, chiamando una routine di Exec: SetIntVector(). Si chiama così:

```
interruptvecchio = SetIntVector (intNum, interrupt)
   D0                           D0-0:4      A1
```

intNum (D0) contiene il numero del bit di interrupt di Paula, di cui volete modificare il vettore: per esempio 3, per Vertical Blank.interrupt (A1) contiene l'indirizzo di una struttura di interrupt server (o nodo) [nota in C come struttura Interrupt, N.d.T.]:

```
STRUCTURE  IS,LN_SIZE

 STRUCTURE  LN,0 ; definizione della struttura List
                Node
   APTR     LN_SUCC  ; = 0
   APTR     LN_PRED  ; = 0
  UBYTE    LN_TYPE  ; = NT_INTERRUPT o NT_SOFTINT
                     (software  interrupt)
   BYTE     LN_PRI   ; = da 0 a 255
   APTR    LN_NAME  ; = indirizzo di una stringa che
                     termina con    uno 0
                   ;per il nome di questo interrupt
   LABEL   LN_SIZE
 APTR   IS_DATA      ; = indirizzo dei dati per questo
                     interrupt
  APTR     IS_CODE   ; = indirizzo della routine di
                     interrupt
 LABEL   IS_SIZE    ; = lunghezza di questa struttura
```

interruptvecchio (D0) è restituito come indirizzo della vecchia struttura server/nodo dell'Interrupt Vector. SetIntVector() può essere chiamata con questo indirizzo in A1 per reinstallare il vettore precedente.
La routine di interrupt che installate con SetIntVector( ), dovrebbe azzerare il bit di richiesta di interrupt in INTREQ ($DFF09C) che ha causato l'interrupt, prima di ritornare.

Per esempio, se avessimo usato SetIntVector( ) per alterare l'Interrupt Vector del Vertical Blank, la nostra routine dovrebbe finire così:

```
MOVE.W #INTF_VERTB, custom+intreq  ;azzera VERTB in
                                   intreq
RTS   ;si ritorna sempre con RTS
```

Per un esempio dell'uso della routine SetIntVector( ), si veda il file SetIntVecEx.asm. Si noti che non dovreste mai alteraredirettamente le strutture Interrupt Vector di Exec, ma usare sempre SetIntVector( ).

Gli interrupt server e le catene di server

In un sistema multitasking come quello di Amiga, è neces-

sario un sistema migliore dei vettori per gestire gli interrupt. Questo avviene perché molti task possono volere che un particolare vettore (come il vettore di Vertical Blank) punti alla propria routine di interrupt. Un vettore può solo puntare ad una routine alla volta, così un task sottrarrebbe il vettore agli altri task, causando il caos.

E' per questo che Amiga supporta le catene di interrupt server. Queste permettono a molti task di condividere lo stesso interrupt, aggiungendo interrupt server a catene di server. Quando si verifica un interrupt, tutti i server di una data catena saranno chiamati uno dopo l'altro, così tutti i task potranno usare lo stesso interrupt.

I principali interrupt che usano catene di server sono l'interrupt Vertical Blank e Copper. Secondo il codice della figura 2, questi interrupt causano un salto alla routine che appare nella figura 3, mediante i loro Interrupt Vector. La routine percorre la catena di interrupt server relativa a questi interrupt, chiamando in maniera ordinata ogni server, finché non viene raggiunta la fine della catena o finché un server non restituisca un valore diverso da zero, che farebbe così terminare la catena.

La routine viene chiamata con A1 che punta alla struttura di intestazione della catena di server. Questa struttura è una struttura privata di Exec [e perciò, come si è detto in precedenza, non si può fare affidamento su di essa]. L'informazione più importante che contiene è costituita dalla prima long word della struttura, che è l'indirizzo della prima struttura di interrupt server della catena.

La routine poi controlla se quel server ha un successore (attraverso il campo LN_SUCC); se non esiste, fa ritorno, altrimenti pone il campo IS_DATA in A1 e il campo IS_CODE in A5 e poi chiama la routine del server.
Al ritorno, se D0 restituisce un valore diverso da zero, la routine esce, altrimenti calcola l'indirizzo del prossimo server della catena, attraverso il campo LN_SUCC e ricomincia il ciclo.
Per aggiungere un interrupt server a una catena, si usi la routine di Exec AddIntServer() in questo modo:

```
AddIntServer(intNum, interrupt)
             d0-0:4  a1
```

intNum (D0) contiene il numero del bit di interrupt di Paula alla cui catena di server volete aggiungere il server.
interrupt (A1) contiene l'indirizzo della vostra struttura per l'interrupt server.Le strutture dei server sono inserite nella catena nel rispetto del valore LN_PRI presente in esse. I server con priorità più alta sono inseriti vicino alla cima della catena e saranno eseguiti per primi.
Il file AddIntSrvEx.asm contiene un esempio che aggiunge un interrupt server alla catena del Vertical Blank per muovere uno sprite sullo schermo.
Quando aggiungete un interrupt server alla catena del Vertical Blank, non usate MAI una priorità maggiore di 10, perché la libreria grafica ha un interrupt server di priorità 10 che ricarica la lista del Copper ad ogni quadro; se viene differita, la lista del Copper non verrà caricata fino al prossimo quadro, cosa che potrebbe generare del caos, specie nel caso in cui delle sezioni grafiche facciano uso della tecnica di double buffering (a doppio buffer).

Le routine degli interrupt server vengono chiamate con le seguenti convenzioni:

D0 - liberamente modificabile
D1 - liberamente modificabile
A0 - liberamente modificabile
A1 - puntatore all'area dati del server (IS_DATA)
(liberamente modificabile)
A5 - liberamente modificabile
A6 - liberamente modificabile

Tutti gli altri registri devono essere preservati. Alcuni documenti indicano che A6 contiene l'indirizzo di ExecBase quando i server vengono chiamati. Questo è corretto solo per il primo server della catena.

Se un server altera A6, il prossimo server riceverà il valore alterato in A6, in quanto la routine in figura 3 non ricarica A6 con il valore di ExecBase, fra una chiamata di un server e la successiva. Così dovreste sempre caricare A6 con ExecBase, se volete accedere ad Exec dai vostri server.

Quando il vostro interrupt server ha terminato, dovrebbe restituire in D0 il valore 0, per permettere che altri server della catena siano eseguiti. Restituire in D0 un valore diverso da zero, farà terminare la catena.

Interrupt software di sistema

Se esaminate le strutture dati che costituiscono la catena di server dell'interrupt Vertical Blank, vi accorgereste che esistono tre server standard. La primo è il server della libreria grafica. Questo essenzialmente si occupa di caricare il Copper con la copperlist corrente, sia per uno schermo non interlacciato (la stessa lista ad ogni quadro), sia per uno interlacciato (alternando fra gb_LOFlist e gb_SHFlist), e di avvisare i task che hanno chiamato WaitTOF().
Il server successivo è quello per il gameport.device: esso legge le gameport dopo un certo numero di quadri.

L'ultimo server della catena è quello per il timer.device: esso calcola il tempo per il timer device VBlank.
La catena di server per il Copper non possiede inizialmente alcun server ed essa viene improvvisamente alla luce solo quando un task gli aggiunge un server e crea una copperlist che causi un interrupt Copper impostando il bit COPER in INTREQ ($DFF09C). L'interrupt Blitter Finished non possiede una catena di server. Invece, chiama una routine che gestisce le richieste di uso del Blitter, prenotate con la routine QBSBlit( ) della libreria grafica.

(segue a pag. 45)

Tre consigli

Il bug di AddSemaphore, il reset legale e un errore abituale con il 68000

Il bug di AddSemaphore

A partire dal momento in cui la Commodore ci ha fornito i semafori per la segnalazione, c'è stato un bug nella routine di Exec AddSemaphore( ) che causa un crash del sistema. Sebbene molti conoscano il bug e un paio di persone abbiano scritto ad Amiga Transactor per segnalarlo, ci sono molti altri che non ne sono a conoscenza (anche se, per essere corretti verso la Commodore, esso sia documentato nel ROM Kernel Manual: Includes & Autodocs v1.3). Pertanto, eccolo qui, per questi ultimi.
Prima di indicare la correzione, stabiliamo in che cosa consista effettivamente l'errore (come esempio interessante di ciò da cui bisogna guardarsi quando si scrive del codice molto ampio). Fondamentalmente, quando si chiama AddSemaphore( ), si corrompe una locazione di memoria posta $10 byte sopra l'indirizzo contenuto in A0.

La funzione AddSemaphore( ) viene chiamata con un puntatore alla vostra struttura signalSemaphore in A1; il formato della funzione AddSemaphore( ) è la seguente:

```
AddSemaphore(signalSemaphore)
             A1
```

La routine dell'OS si comporta a questo modo (gli estratti dalla ROM sono tratti dalla versione 1.2, v33.5 Exec 33.192):

```
FC2F24  jsr   -$022E(A6)   ; jsr InitSemaphore()
FC2F28  lea   $0214(A6),(A0) ; ptr alla lista globale
                             dei semafori
FC2F2C  bra   $FC1682       ; Enqueue protetto, che
                             aggiunge il
                            ; semaforo alla lista
```

Ora... giunti a InitSemaphore(), il problema ci si presenta sin dalla prima linea di codice:

```
FC2D94  lea     $10(A0),A1 ; punta alla lista dei
                             task in attesa
FC2D98  move.l  A1,(A1)   ; la inizializza come vuota
FC2D9A  addq.l  #4A1
FC2D9C  clr.l   4(A1)
FC2DA0  move.l  A18(A1)
FC2DA4  clr.l   $28(A0)         ; imposta ss_fields
FC2DA8  clr.w   $0E(A0)
FC2DAC  move.w  #$FFFF,$2C(A0)
FC2DB2  rts
```

Da ciò si può vedere che InitSemaphore( ) si aspetta che il puntatore a signalSemaphore sia in A0, proprio come stabilisce il formato della funzione:

```
InitSemaphore(signalSemaphore)
              A0
```

Ma noi l'abbiamo posto in A1 quando abbiamo chiamato AddSemaphore( ) e là si trova; come risultato, A0 potrebbe contenere qualsiasi cosa quando si chiama $FC2D94; LEA carica un indirizzo posto $10 byte sopra (A0) in A1 e l'istruzione seguente ne corrompe il contenuto!
E non basta, la routine AddSemaphore( ) assume che A1 sia intatto, al ritorno da InitSemaphore( ). Nessuna meraviglia che non funzioni. La Commodore ha offerto la seguente soluzione al problema:

```
#include "exec/execbase.h"
...
void  AddSemaphore(s)

struct SignalSemaphore *s;
{
  InitSemaphore(s);
  Forbid();
  Enqueue(&SysBase->SemaphoreList,s);
  Permit();
}
```

Di fatto questa correzione è solo una versione C di ciò che il codice sorgente originale tentava di fare in qualche modo.

Reset: il modo appropriato

Sebbene sia un caso raro, un programmatore può volere che Amiga faccia un boot sotto controllo software. Per far questo, i programmatori sono stati molto intraprendenti e hanno inventato molteplici schemi, ma molti di questi si sono dimostrati fallimentari con CPU o configurazioni di memoria differenti.
Quello che segue è il metodo ufficiale della Commodore

per mantenere la compatibilità in tutti i sistemi:

```
   XDEF     _ColdReboot
   XREF     _LVOSupervisor

_ColdReboot:
   move.l  4,a6              ; ricava ptr a ExecBase
   lea.l   Resetcode(pc),a5  ; indirizzo del codica da
                               lanciare
   jsr     _LVOSupervisor(a6) ; lo lancia in modo
                                  supervisore

  CNOP   0,4 ;il codice di reset deve essere allineato
           alla longword

ResetCode:
   lea.l   2,a0      ; punta a JMP all'inizio della ROM
   RESET             ; fa scomparire la RAM
   jmp    (a0)       ; e salta nella ROM

   END
```

ResetCode si fonda sul fatto che l'istruzione JMP(A0) viene pre-caricata. Se così non fosse, la RAM scomparirebbe e non si potrebbe mai vedere l'istruzione JMP! ResetCode deve essere esattamente come è stato specificato, altrimenti il codice potrebbe non funzionare correttamente in qualche configurazione del sistema.

Un errore col 68000

C'è una piccola peculiarità nella famiglia di processori del 68000 che inganna molti programmatori. E' facile caderci, perché non è sempre chiaro nella documentazione: le istruzioni del 68000 non influenzano i codici di condizione del processore quando il registro destinazione è un registro indirizzi.
Come esempio, prendiamo il codice di Jim Butterflied in Transactor per Amiga, n.6, 1989 p.52:

```
 MOVE.L  $4,A6            ; base dell'Exec
 LEA     DosName(PC),A1   ; puntatore nome DOS
 MOVEQ   #0,D0            ; qualsiasi versione
 JSR     _LVOOpenLibrary(A6)
 MOVE.L  D0,A6            ; puntatore base DOS
 BEQ.S   Exit             ; zero, esci
```

Al ritorno da OpenLibrary(), D0 contiene il puntatore alla libreria DOS. Jim lo sposta dove è necessario (in A6) e assume che questo modifichi i flag di condizione, così che il seguente BEQ causi un salto se la libreria DOS non si fosse aperta.
Tuttavia lo spostamento di D0 in A6 non ha impostato i flag di condizione; ciononostante, dal momento che è estremamente raro che la libreria DOS non si apra, il salto non sarebbe normalmente effettuato e il codice funzionerebbe ancora.
Non è solo Jim a commettere questo errore: sembra essere un errore piuttosto comune fra i programmatori più esperti (e Jim deve essere annoverato fra questi!).
Così, si ricordi, ogni operazione che invia il risultato in un registro indirizzi, non influirà sui flag di condizione.

Si ringraziano R.J.Appleton, Bryce Nesbitt e Luc Van Rompaey per le informazioni.

(segue da pag. 43)

Gli interrupt di Amiga

Gli Interrupt Vector dei canali audio non vengono abilitati finché o il device audio o qualche programma non esegua per essi la routine SetIntVector( ). Si veda il file SetIntVecEx.asm per un esempio.

Gli interrupt Disk Block Finished (conclusione di blocco del disco) e Disk Sync saltano a routine nel trackdisk.device che controllano la lettura e la lettura dei dischi.
Il Software Interrupt è uno speciale interrupt che può essere generato da un programma impostando il bit SOFT in INTREQ ($DFF09C) o chiamando la routine Exec Cause( ), che causerà il Software Interrupt.

Mettiamo tutto in pratica

Bene, ora che abbiamo esaminato la maggior parte della teoria sugli interrupt di Amiga, daremo uno sguardo al modo con cui possiamo usarli effettivamente nei nostri programmi. Il file AddIntSrvEx.asm mostra come aggiungere un interrupt server alla catena dei server del Vertical Blank. Il programma è scritto in Assembler perché questo è il linguaggio migliore da usarsi con gli interrupt, a motivo della sua velocità.

Il file SetIntVecEx.asm dimostra come alterare gli Interrupt Vector di Exec. Gli Interrupt Vector per i canali audio non sono impostati adeguatamente su Amiga finché non viene usato il device audio. Pertanto, il programma crea un vettore la cui routine per disabilitare il suono DMA di un canale audio, è localizzata nel proprio codice.

Entrambi i listati sono stati scritti con DevPac V2.08 della HiSoft e dovrebbero fornirvi una base utile per scrivere le vostre routine di interrupt.
Divertitevi!

Le librerie residenti dell'AmigaDOS

L'hunk che il tempo (e la maggior parte di noi) ha dimenticato

Eric Salter

Il problema

C'è molto nell'AmigaDOS che salta agli occhi. Non solo è una delle librerie più affascinanti del sistema operativo di Amiga (è scritta in BCPL, un linguaggio sconosciuto alla maggior parte di noi), ma ha anche alcune caratteristiche molto interessanti che sono a mala pena menzionate nella documentazione del sistema. Questo articolo tratta di una di queste caratteristiche: il supporto per le librerie residenti del loader dell'AmigaDOS.

Librerie, librerie e "librerie"

Che cos'è una libreria? La risposta dipende da quale area del software di sistema ci si stia occupando.
Le librerie stile Exec sono ben note a noi tutti: la loro struttura Library è collegata alla lista delle librerie di Exec durante il loro processo di inizializzazione. Noi apriamo queste librerie chiamando la funzione Exec OpenLibrary( ) e i loro servizi sono accessibili saltando a un offset negativo rispetto all'indirizzo della base della libreria che ci è stato restituito da OpenLibrary( ). L'offset per un particolare servizio della libreria rimane immutato fra una release del software di sistema e la successiva.

Le librerie linked sono familiari a coloro che scrivono il proprio codice. Sono le librerie, come l'amiga.lib e la lc.lib, attraverso le quali i riferimenti irrisolti alle routine o ad altri dati simbolici presenti nel nostro codice C (o di altri linguaggi di alto livello) vengono finalmente risolti, incorporando moduli di codice da queste librerie per soddisfare il linker.
La terza classe di librerie nel sistema Amiga è poco conosciuta, o, piuttosto, conosciuta in una forma più tradizionale. Sono le librerie residenti dell'AmigaDOS che sono state documentate in maniera povera, menzionate raramente e hanno languito nell'oscurità dall'inizio dei secoli, ma che offrono alcune caratteristiche che meritano di essere esplorate ulteriormente dalla comunità di Amiga.

E' noto che alla Commodore stanno pensando di rimuovere dal loader del DOS il codice di supporto per il link dinamico delle librerie residenti che entra in funzione in load-time (fase di caricamento). Ho scritto questo articolo per documentare completamente questa caratteristica e dimostrare che si tratta di un'utile aggiunta agli strumenti di lavoro del programmatore.
Una domanda di grazia alla Commodore: per favore non rimuovetelo, ciò distruggerebbe il mio codice!

Il link delle librerie residenti in load-time

Il loader del DOS è in grado di aprire automaticamente le librerie di sistema, attraverso OpenLibrary( ) e di rilocare i *riferimenti a queste librerie, in load-time.*
La pratica corrente fra i programmatori di linguaggi di alto livello e assembler è quella di avere del codice di startup standard e ben collaudato, che apra le librerie Exec, DOS e Intuition e ponga l'indirizzo della loro base in variabili statiche globali.

Con il loader del DOS è possibile che un programma specifichi al loader da quali librerie dipende per le proprie operazioni. Il loader aprirà queste librerie per il programma e poi rilocherà i riferimenti a queste librerie durante il caricamento. Per esempio, il codice assembly per chiamare la funzione Read() della libreria DOS potrebbe essere scritto:

```
TheRead: movea.l  DOSBase, A6 ; carica il DOS
         jsr      _LVORead(A6) ; chiama la routine
         ...      ...
         resto del codice
```

senza avere esplicitamente aperto la libreria DOS per ricavare DOSBase. Alternativamente, è possibile che il loader corregga i riferimenti in modo che solo:

```
         jsr      Read
```

sia necessario per chiamare la funzione Read( ) del DOS, ancora una volta senza avere aperto formalmente la libreria. Ecco alcuni dei maggiori vantaggi che derivano da questa tecnica:

* il codice è caricato solo se le librerie sono presenti
* le librerie vengono aperte e chiuse automaticamente
* il codice sorgente e quello eseguibile sono più brevi
* i salti sono leggermente più veloci, assoluti e non indicizzati
* possibilità di link in run-time di programmi differenti, non

solo di moduli entro un singolo programma

Ci sono tuttavia alcuni svantaggi:

* blink non supporta le librerie residenti: solo Alink le permette, per il momento
* la Commodore può rimuovere il supporto per il loader: è l'ostacolo maggiore
* non c'è alcuna possibilità di avvertire l'utente di quale libreria manca (solo un limpido ERROR 122: libreria residente non valida durante il caricamento)
* non si può specificare una particolare versione della libreria da aprire
* la fase di link è leggermente più complessa

Come è possibile? Nell'AmigaDOS Technical References Manual, compare un succinto riferimento al processo. Leggiamo:

"I file caricabili sono anche noti come "librerie"... E' possibile riferirsi alle librerie residenti attraverso riferimenti esterni; le definizioni sono in un hunk che non contiene codice, ma solo una lista delle definizioni della libreria residente.
Normalmente, per produrre questi hunk, si assembla un file che contiene solo definizioni esterne assolute e poi lo si fa passare attraverso un programma speciale per convertire le definizioni assolute in definizioni della libreria residente. Il linker usa il nome dell'hunk come nome della libreria residente e lo ripone nel file caricabile in modo che il loader possa aprire la libreria residente prima dell'uso."

... e questa è quasi tutta l'informazione di cui disponiamo. Che cos'è questo "programma speciale" e dove trovarlo? Bene, non sono stato capace di trovarlo (non ho cercato molto), ma si continui a leggere per avere una soluzione al problema (e si veda il listato di mkres su disco).

La creazione delle definizioni della libreria residente

Per creare la definizione di una libreria residente, dapprima creiamo un modulo di riferimenti esterni.
Ecco una semplice definizione per alcune routine della libreria DOS:

```
 SECTION dos.library,DATA

 * creiamo gli offset
   DOSBase  EQU   0
   Open     EQU   -30
   Close    EQU   Open-6
   Read     EQU   Close-6
   Write    EQU   Read-6
   Input    EQU   Write-6
   Output   EQU   Input-6

* prepariamo le definizioni esterne
        XDEF     DOSBase
        XDEF     Open
        XDEF     Close
        XDEF     Read
        XDEF     Write
        XDEF     Input
        XDEF     Output

        END
```

Quando sarà assemblato, il file oggetto conterrà una serie di definizioni assolute per i simboli che abbiamo definito. Poi facciamo passare il file attraverso il nostro convertitore di dati per trasformare le definizioni assolute in definizioni della libreria residente. Il codice citato genera il seguente codice oggetto:

```
0000: 000003E7 00000000 000003E8 00000003   ...?.......?....
0010: 646F732E 6C696272 61727900 000003EA   dos.library....?
0020: 00000000 000003EF 02000002 4F757470   .......?....Outp
0030: 75740000 FFFFFFC4 02000002 496E7075   ut..????....Inpu
0040: 74000000 FFFFFFCA 02000002 57726974   t...????....Writ
0050: 65000000 FFFFFFD0 02000001 52656164   e...????....Read
0060: FFFFFFD6 02000002 436C6F73 65000000   ????....Close...
0070: FFFFFFDC 02000001 4F70656E FFFFFFE2   ????....Open????
0080: 02000002 444F5342 61736500 00000000   ....DOSBase.....
0090: 00000000 000003F2                     .......?
```

Questo codice contiene:

```
hunk_unit     con nessun campo nome
hunk_name     "dos.library"
hunk_data     con nessun dato
hunk_ext      con 7 definizioni esterne assolute:
              Output  valore FFFFFFC4
              Input   valore FFFFFFCA
              Write   valore FFFFFFD0
              Read    valore FFFFFFD6
              Close   valore FFFFFFDC
              Open    valore FFFFFFE2
              DOSBase valore 0
hunk_end
```

Dopo aver fatto passare questo file oggetto attraverso mkres, il convertitore di dati che accompagna questo articolo, il codice appare nel seguente modo:

```
0000: 000003E7 00000000 000003E8 00000003   ...?.......?....
0010: 646F732E 6C696272 61727900 000003E6   dos.library....?
0020: 000003EF 03000002 4F757470 75740000   ...?....Output..
0030: FFFFFFC4 03000002 496E7075 74000000   ????....Input...
0040: FFFFFFCA 03000002 57726974 65000000   ????....Write...
0050: FFFFFFD0 03000001 52656164 FFFFFFD6   ????....Read????
0060: 03000002 436C6F73 65000000 FFFFFFDC   ....Close...????
0070: 03000001 4F70656E FFFFFFE2 03000002   ....Open????....
0080: 444F5342 61736500 00000000 00000000   DOSBase.........
0090: 000003F2                              ...?
```

che contiene:

```
 hunk_unit     con nessun campo nome
 hunk_name     "dos.library"
 hunk_resident un hunk non documentato in precedenza
 hunk_ext     con 7 definizioni di libreria residen-
te:
              Output  offset -0x3C
              Input   offset -0x36
              Write   offset -0x30
              Read    offset -0x2A
              Close   offset -0x24
              Open    offset -0x1E
              DOSBase offset  0
 hunk_end
```

Ora eseguiremo il link di questo hunk con un modulo di codice, una versione assembler dell'immortale classico di Kernighan e Ritchie:

```
IDNT   Hello_World

XREF   Open
XREF   Write
XREF   Output

SECTION text,CODE

START  jsr     Output  ;ricava canale corrente di
                        output
       tst.l   d0      ; controllo di errore
       beq.s   quit    ; esce in caso di errore
       move.l  d0,d1   ; sposta file handle
       lea     Hello(pc),a0 ; pone la stringa col
                              messaggio...
       move.l  a0,d2   ; ...nel puntatore al buffer
       moveq.l #size-Hello,d3 ; ricava lunghezza
       jsr     Write   ; chiama il DOS

quit   rts

Hello  DC.B    'Hello,World',$A
size

END
```

Se osserviamo il disassemblato del codice, dopo che è stato caricato dal loader del DOS, apparirà a questo modo:

```
C76BC8 jsr     $C04B94.l
C76BCE tst.l   d0
C76BD0 beq.s   $C76BE2
C76BD2 move.l  d0,d1
C76BD4 lea.l   $E(PC),a0
C76BD8 move.l  a0,d2
C76BDA moveq.l #$D,d3
C76BDC jsr     $C04BA0.l
C76BE2 rts
C76BE4 46F2C20576F726C640A Hello, World^J
```

Quando il programma è stato caricato, il nodo della libreria DOS risiedeva all'indirizzo $C04BD0, così gli indirizzi per le istruzioni JSR sono corrette! Come funziona questa operazione di rilocazione in load-time?

Il metodo usato

Del frammento di codice più sopra, solo il sorgente delle definizioni esterne è stato assemblato con un assembler compatibile con quello della Metacomco. Gli altri sono stati assemblati con il Lattice ASM perché è più veloce e, in generale, è un assembler migliore.
Il programma ASM della Lattice non obbedisce alle regole; esso genera ext_hunknon standard che definiscono valori esterni assoluti, aggiungendo il prefisso _Abs all'hunk e impostando il tipo EXT, in maniera scorretta, a 0x01: perciò non può essere usato in questo caso (non ho niente contro la Lattice, sono un utente registrato del Lattice 5.02 e ne sono molto contento).

Il programma originale Alink è stato usato per effettuare il link del codice oggetto, al fine di produrre il file eseguibile e caricabile. Blink non può generare hunk del tipo hunk_header nel file caricabile con riferimenti alle librerie residenti e tanto meno far fronte al tipo di hunk non documentato, hunk_resident, del codice oggetto. Non

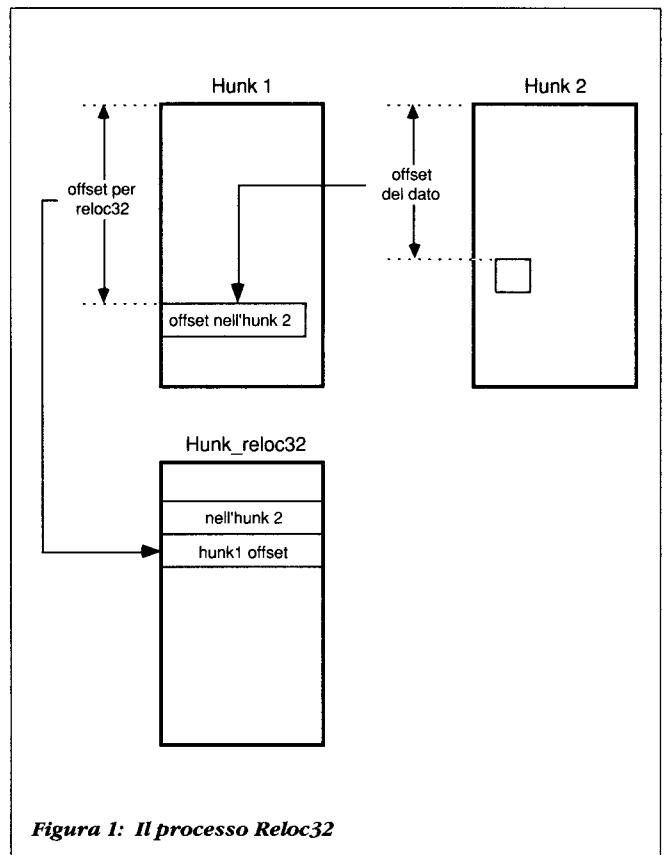



Figura 1: Il processo Reloc32

posso fare commenti sull'adeguatezza del Manx Aztec, ma sospetto che sia dello stesso livello del Lattice.
Le seguenti linee di comando eseguono la compilazione e il link:

```
assem   dos.asm -o dos.o ; la libreria residente
asm     hello.asm       ; il codice di "Hello World"
mkres   dos.o resdos.o   ; trasforma il modulo
alink   resdos.o+hello.o TO Hello
                         ; l'eseguibile finale
```

Che cos'è il link in load-time ?

Il link in load-time o in run-time (fase di lancio) delle librerie residenti è un'estensione della già familiare rilocazione dinamica dei valori a 32 bit, che avviene tra il codice e l'area dei dati in load-time. Se un programma consiste di moduli multipli, il codice entro un modulo può fare riferimento al codice o ai dati presenti in un altro o anche a locazioni entro se stesso. Siccome il codice può essere caricato in maniera discontinua (scatter-loaded) in qualsiasi punto della memoria dal loader del DOS, il codice non può sapere, prima di essere effettivamente caricato, dove le varie parti di se stesso risiedono fisicamente nella memoria di Amiga. A questo fine, ci sono informazioni di rilocazione nel file caricabile che identificano le locazione che devono essere "riparate" per riflettere l'indirizzo finale del codice.

Con riferimento alla figura 1, un diagramma esemplificativo del processo reloc32, abbiamo una locazione nell'hunk 1 che si riferisce alla locazione di un byte nell'hunk 2.
Durante il processo di link, l'offset di quel byte a partire dall'inizio dell'hunk 2 viene calcolato e immagazzinato nella longword di rilocazione nell'hunk 1. Si noti che la locazione di questa longword nell'hunk 1 dovrà essere rilocata in load-time, così viene creato un elemento nella tavola di rilocazione dell'hunk 1. La tavola di rilocazione specifica che la longword al nostro offset deve essere rilocata in rapporto all'hunk 2 quando verrà caricato. Il lavoro viene eseguito in load-time, quando viene allocata la memoria per gli hunk; il loader aggiunge l'offset immagazzinato nell'hunk 1 all'indirizzo della base dell'hunk 2, risolvendo così in maniera compiuta il riferimento.

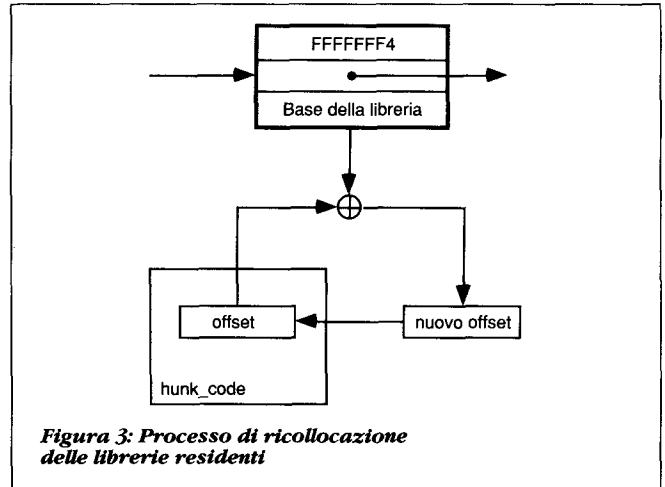



Figura 3: Processo di ricollocazione delle librerie residenti

Il loader del DOS e le librerie residenti

Quando il DOS carica del codice, il loader alloca memoria per ogni singolo hunk di tipo CODE, DATA o BSS presente nel file caricabile.
Queste regioni di memoria allocate sono dette "segmenti" nel gergo dell'AmigaDOS. Un BPTR (puntatore BCPL) è restituito dalla funzione DOS LoadSeg( ). Questo puntatore (dopo la conversione) punta a una lista semplice di segmenti che costituiscono il programma. L'esecuzione del codice viene compiuta saltando a questo indirizzo + 4 byte.
Nei programmi che approfittano della capacità di gestire librerie residenti del loader, il primo hunk che si incontra nel file caricabile, l'hunk_header, contiene la lista delle librerie, nell'ordine in cui devono essere aperte dal loader.
Queste librerie vengono aperte mediante una chiamata alla funzione Exec OpenLibrary( ) e nel caso in cui vengano aperte con successo, si accoda uno speciale segmento alla

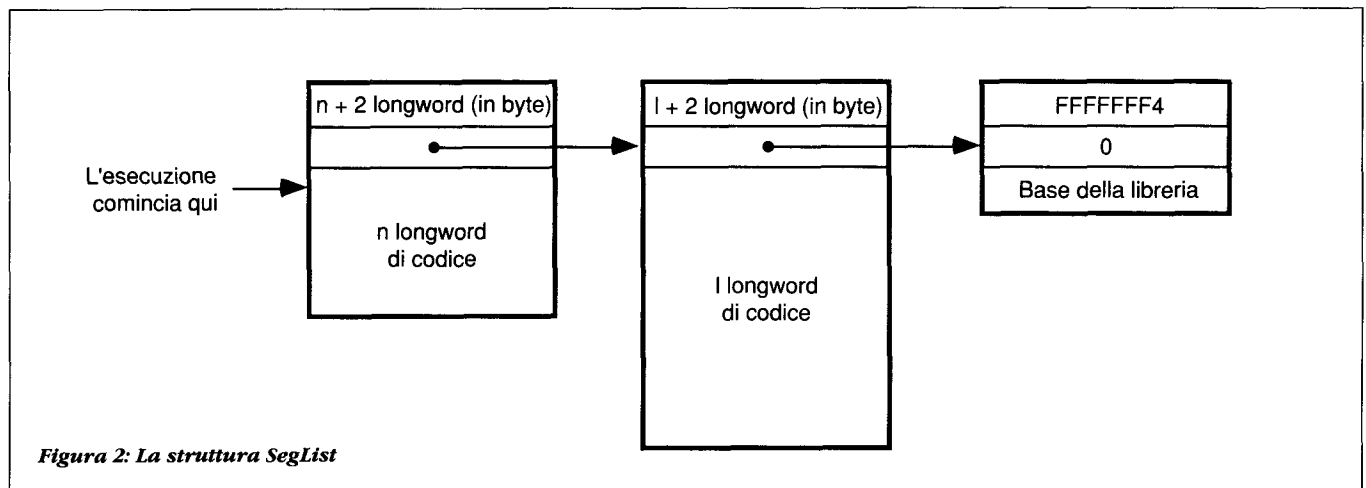



Figura 2: La struttura SegList

lista, completa solo in parte, dei segmenti. Questo processo avviene prima di caricare il codice così quel tempo non viene perduto se una libreria non si dovesse aprire. Dopo il caricamento, la SegList appare come quella della figura 2.

Un segmento di libreria residente è una struttura di 12 byte con l'indirizzo della base del nodo di libreria, quello ritornato da OpenLibrary(), posto all'indirizzo del segmento + 4 e la lunghezza del segmento (12 byte) posta all'indirizzo del segmento - 4 nella forma del complemento a uno.
La rilocazione di una longword, in un hunk che si riferisca a una libreria residente, avviene come se la libreria fosse proprio un altro hunk; solo accade che l'indirizzo della sua base sia quello della base di un nodo di libreria. Questo è rappresentato graficamente nella figura 3.

Attenzione

Il lettore astuto avrà notato qualcosa di strano in "Hello World".
Ci si aspetta che tutte le librerie siano chiamate con il loro indirizzo base nel registro A6, ma, nell'esempio precedente, DOSBase non viene caricato in alcun registro.
La libreria del DOS costituisce l'eccezione a questa regola; è scritta in BCPL e l'interfaccia finale della libreria, in stile Exec, non richiede che questa convenzione sia rispettata. Ciò non preclude l'uso della caratteristica delle librerie residenti con altre librerie, in quanto possiamo caricare esplicitamente A6 con la base della libreria usando la stessa tecnica di prima e chiamare ancora le routine con riferimenti assoluti:

```
movea   #IntuitionBase,A6
jsr     OpenWindow
```

Come funziona mkres

Il programma mkres è la chiave per rendere disponibili al linker le giuste definizioni delle librerie residenti. mkres converte il codice oggetto dell'assembler in qualcosa che appare in maniera simile a questa:

hunk_unit nome opzionale
hunk_name nome della libreria nel file caricabile (si veda la figura 4)
hunk_resident
nx hunk_ext (si veda la figura 5)
hunk_end
... qualsiasi numero di ripetizioni di questo gruppo, per ciascuna libreria definita.

Come apparirà il file caricabile finale? Ecco finalmente l'output della fase di link di "Hello World":

```
0000: 000003F3 00000003 646F732E 6C696272 ...?....dos.libr
0010: 61727900 00000000 00000001 00000001 ary............
0020: 00000001 0000000B 000003E9 0000000B ...........?....
0030: 4EB9FFFF FFC44A80 67102200 41FA000E N?????J.g.".A?..
0040: 2408760D 4EB9FFFF FFD04E75 48656C6C $.v.N?????NuHell
0050: 6F2C2057 6F726C64 0A000000 000003EC o, World.......?
0060: 00000002 00000000 00000002 00000016 ................
0070: 00000000 000003F2                   .......?
```

Stiamo vedendo un file caricabile che, probabilmente, la maggior parte dei programmatori di Amiga non ha mai visto prima e forse il più breve "Hello World" che può essere scritto su Amiga! Io spero che ciò sia stato interessante; scommetto che voi non sapevate che Amiga aveva questa capacità. E' stato comunque interessante per me, amo risolvere misteri, e uno dei misteri di Amiga che mi ha infastidito per un certo tempo è finalmente caduto grazie un po' di lavoro di investigazione.

Non pensiate che questa sia la parola conclusiva sul soggetto; ci sono molte possibilità che non ho esplorato qui, come scrivere delle librerie che approfittino maggiormente di questo processo di link dinamico in run-time. Un pensiero finale sul link run-time: l'AmigaDOS lo possedeva molto prima dell'OS/2; forse possiamo riscoprirne l'utilità.

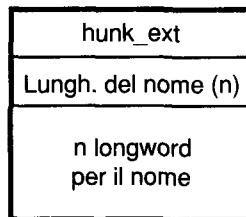



Figura 4: La struttura hunk_name

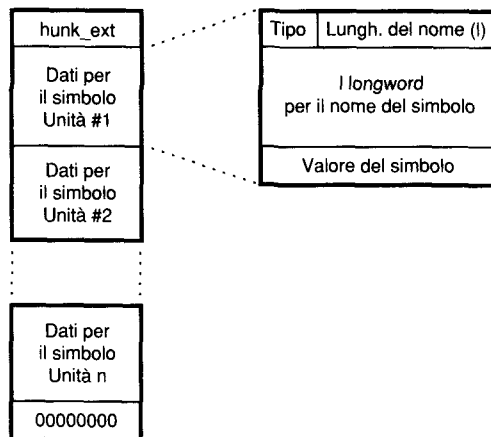
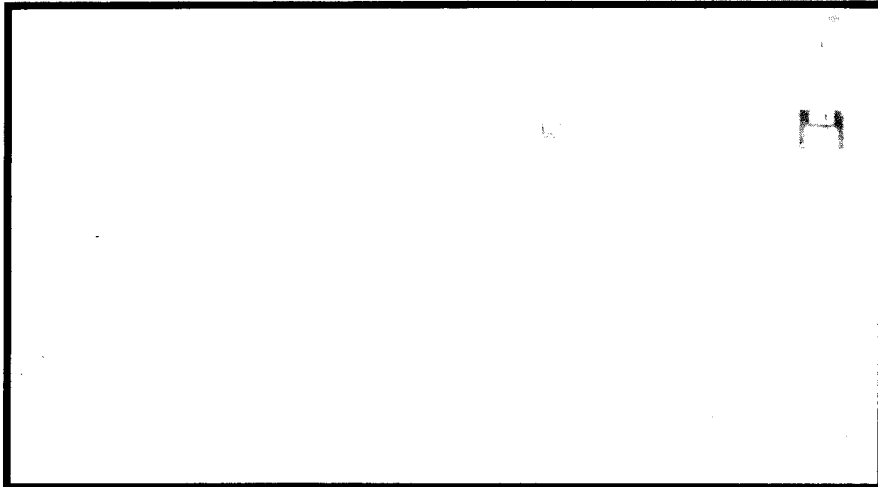



Figura 5: La struttura hunk_ext

bootblock del dischetto che si vuole installare con StealMemBoot, in quanto l'utility non controlla se il bootblock precedente era o no standard; può infatti accadere che modificando il bootblock il programma non funzioni più, e questo perché nel bootblock era presente un programma che dava inizio al caricamento di altre parti del gioco (come accade ad esempio per i giochi della Psignosys).
Quindi il Gruppo Editoriale Jackson NON si assume alcuna responsabilità per eventuali danni provocati a dati o programmi a causa di un uso incauto di StealMemBoot.

PopInfo

PopInfo è un'utility che permette di ottenere un grande numero di informazioni sul il sistema, con più facilità e chiarezza rispetto a quanto non faccia l'opzione Info del menu del WorkBench.
Facendo girare il programma, non accade nulla, se non la trasformazione del Wordbench in un Porkbench (questo perché una piccola finestrella con impressa una P viene posizionata nell'angolo superiore-sinistro dello schermo, sulla dragbar). Per attivare PopInfo basta cliccare su quella P, e sullo schermo si aprirà una finestra contenente, per ogni device attivo, le seguenti informazioni:

Unit Bytes Used Free Ers WPS SBB

Eccone il significato:

- **UNIT**
E' il nome del device (per esempio DF0:, RAM:, VD0: eccetera)
- **BYTES**
E' il numero totale di Byte disponibili su quel device (molto più utile che il numero di blocchi totale, come indicato da Info del Workbench)
- **USED**
E' il numero di Byte utilizzati su quel device
- **FREE**
E' il numero di Byte liberi su quel device
- **ERS**
E' il numero di errori non eliminabili (Hard Errors) su quel device.
- **WPS**
E' lo stato di quel device (R/W in caso si possa leggere e scrivere, R O nel caso si possa solo leggere, oppure N/A per device che non hanno un particolare stato di protezione).
- **SBB**
Sotto questa voce viene indicato se il device ha un bootblock standard (YES), non standard (NO) o se manca di bootblock (NBB).
Per device in cui il controllo del bootblock non ha senso viene indicato N/A.

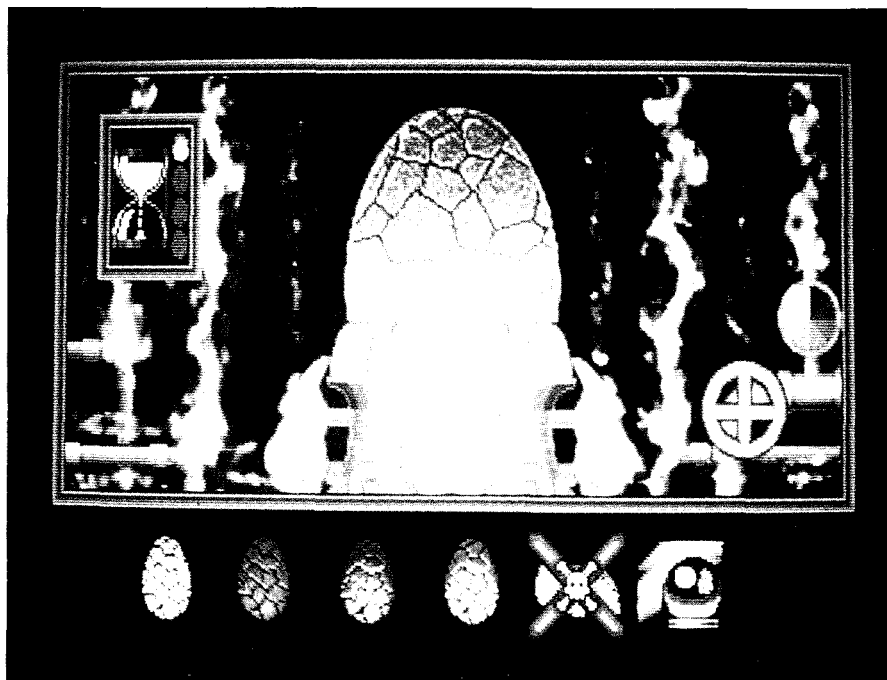
Se in un particolare drive non è presente il disco, questo viene indicato con la dicitura "No disk present in drive". Se, invece, il disco non è DOS apparirà il messaggio "Not a DOS disk".
Vengono, inoltre, visualizzate informazioni relative alla CHIP RAM libera, alla FAST RAM libera, alla RAM TOTALE libera e all'orario corrente.
Se PopInfo trova un Bootblock non standard, vi permetterà di vederne il contenuto ed eventualmente di installare con un bootblock standard il disco. Per disattivare PopInfo basta cliccare all'esterno della finestra, ed essa si rimpicciolirà nuovamente riportandosi alle dimensioni del piccolo rettangolino con la P.

• Scherzi

Avvertenze per l'uso dei programmi-scherzo

I due programmi-scherzo presenti nel drawer scherzi del disco di questo mese, sono programmi che, solitamente, girano in background, impegnando quindi parte del tempo macchina messo a disposizione dal microprocessore e spesso utilizzando, senza troppe precauzioni, l'hardware di Amiga. Può quindi accadere, facendone girare più di uno alla volta, che il sistema possa andare in crash. Se dovesse succedere, limitatevi ad utilizzare solo uno scherzo alla volta e non avrete problemi (anche se a volte, ma più raramente, può accadere che uno scherzo entri in conflitto anche con un normale programma, presente negli altri drawer).

Dragons Breath

In questo numero di Amiga Magazine siamo orgogliosi di offrire ai nostri affezionati lettori una stupenda demo dell'ultima fatica della Palace Software: il famoso Dragons Breath!

La storia

Dragons Breath è un gioco d'azione con elementi sia di strategia che di gioco di ruolo. Lo scenario fantastico delle Terre di Anrea costituisce l'immensa area d'azione per i tre partecipanti il cui fine è impossessarsi del segreto dell'immortalità ben custodito in un quasi inaccessibili Castello. Il computer può partecipare attivamente assumendo uno o due ruoli, o giocando da sé gestendo tutti i tre i contendenti!
Dragons Breath contempla innumerevoli fasi il cui sviluppo può determinare il successo o meno della partita; ognuno, infatti, deve riempire informazioni, effettuare acquisti dai mercanti di passaggio, avere una rudimentale conoscenza dell'alchimia per poterne fare uso al momento più opportuno. La conquista dei villaggi e la successiva protezione del proprio dominio è delegata a un'originale formazione bellica costituita da draghi che si devono allevare già dallo stadio di uova di drago. Dragons Breath è dunque un programma ricreativo che non si brucia in poche sedute: a seconda delle attitudini dei giocatori e grazie alle possibilità di memorizzare ogni punto della partita, il gioco può durare quasi all'infinito.
Dragons Breath è il primo prodotto Palace realizzato completamente in italiano: sia il testo nel gioco che l'ampio manuale di istruzioni sono complertamente in italiano. Anche il misterioso libro dei sortilegi, incluso nella confezione, è stato tradotto in italiano. L'aspetto più appariscente e convincente di Dragons Breath è senza dubbio l'ottima qualità della grafica e dell' animazione.

CeBIT 1990: Amiga, Amiga e ancora Amiga.

Carlo Cattoni

Una vera "abbuffata" di applicazioni per il nostro beneamato computer: questo è quanto abbiamo potuto osservare al CeBIT, annuale kermesse "computereccia" che si tiene a Marzo negli enormi quartieri fieristici di Hannover, in Germania.
Lo stand Commodore, come al solito uno dei più affollati di tutta la fiera, era organizzato in modo da distinguere nettamente le aree dedicate ai PC e quelle dedicate ad Amiga; le zone dedicate a quest'ultimo computer erano a loro volta suddivise in applicazioni professionali e di home office.
Gli espositori tedeschi erano forniti di A2500/30, macchine che in Italia non sono mai state commercializzate, ma che comunque vantano prestazioni notevoli (e si poteva toccare "con mano" durante le dimostrazioni di ray-tracing...).

Le novità

La novità più interressante, secondo noi, è stata presentata da una ditta tedesca indipendente.
Una scatoletta di espansione che conteneva "solo" una scheda Transputer e un scheda video a trentadue bit e sedici milioni di colori !!

La Gold Disk, la famossissima ditta Canadese, dimostrava il suo potentissimo programma per DeskTop Publishing: Professional Page 1.3; inoltre presentava una beta-version di Professional Draw 2.0, un super programma da far "strabuzzare gli occhi" agli utenti di Mac e PC.
In un'isola era seminascosto un VAX, al quale era magnificamente accoppiato un Amiga come terminale grafico; su Amiga girava XWindows.
Naturalmente le applicazioni MIDI non potevano mancare.
La Kawai presentava un package con tastiera polifonica amplificata e software sviluppato dalla Steimberg. Per il resto abbiamo potuto notare le registrazioni di applicazioni grafiche, molto belle, realizzate con i vari Sculpt 4D e Caligari; delle dimostrazioni di home accounting, e didattiche, realizzate con linguaggi "classici" e con i recenti Can Do, Deluxe Video III, Elan Performer, Viva, Authoring System ecc.
Infine abbiamo notato un genlock con caratteristiche molto interessanti: effetti di wiping circolari e rettilinei con temporizzazioni regolabili (una cosa simile era prerogativa assoluta del Magni) e uscita su videoregistrazione anche in Super VHS.

Gli altri

La Combitec presentava Medusa, emulatore di Atari ST funzionate, per ora, al 90%, la Epson una stampante a 64 aghi, la Tektronix le nuove stampanti a colori Postscript (già viste alla I.CO.Graphics) a trasferimento termico, la Vortex i famosissimi hard disk e la Sharp i nuovi scanner; infine, le applicazioni video di Amiga erano esposte nella sala dedicata al computer del 2000.
Per concludere una curiosità, una ditta americana presentava una stampante inkjet che disegna poster di circa nove metri quadri (3x3 m), il costo ci sembrava veramente alla portata di tutti: circa 200 milioni di lire !!!! ▲

Articoli tradotto dal n.3 Fall 1989 di
Compute!'s Amiga Resource.

Pensare in grande, a grandi giochi

Come programmare un grande gioco per Amiga ? La risposta è complessa, ma c'è chi se la sente di affrontare questa sfida. Abbiamo parlato con alcuni di quelli che pensano di conoscere le tecniche più avanzate.

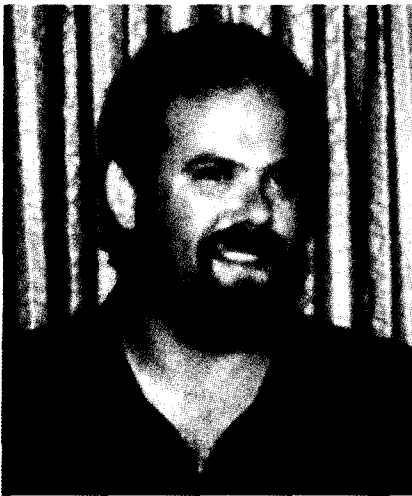
Keith Ferrell

Parlando di Amiga come macchina per giocare, non possono che suscitare consensi. Sicuramente è un'ottima macchina per giocare, ma è altrettanto vero che i giochi migliori devono ancora essere sviluppati.
Ci sono molte spiegazioni a questo fatto, tante quante sono gli sviluppatori e gli editori di giochi che supportano Amiga. Per scoprire cosa rende Amiga così speciale e cosa rende speciale un gioco per Amiga, ci siamo rivolti ad alcuni dei leader, e ad alcuni membri dell'ultimissima leva di programmatori, nel settore del software ludico.

Discovery Software
Progettare sinfonie interattive

Quando pensa all'intrattenimento su Amiga, Rick Ross pensa in termini musicali.
Presidente della Discovery Software, editrice di Sword of Sodan e Arkanoid, Ross vede il software ricreativo come una sinfonia e Amiga come un'orchestra, diretta da programmatori e progettisti.
"Il meccanismo centrale del gioco deve funzionare esattamente come un grande direttore conduce l'orchestra" dice. "Bisogna chiamare in causa certe risorse esattamente al momento giusto, dosare la potenza del computer, dirgli quando fermarsi per un attimo, e così via".
Diversamente dal mondo musicale, gli strumenti di un programmatore sono in continua evoluzione. "La tecnologia è un bersaglio in continuo allontanamento, le mete di oggi sono la storia di domani".
Allora, come se la cava Amiga nel confronto con le ultimissime macchine per giocare?
"Ai tempi della sua ideazione" dice Ross "Amiga era la macchina ideale per giocare con la tecnologia allora a disposizione. Lo stato dell'arte si è evoluto, e oggi compaiono nuove console, affascinanti strumenti di seconda generazione. Ma perfino a confronto con la migliore console, Amiga esce sempre vincente".
Perché?
"Perché Amiga offre una gran quantità di opzioni che permettono la manipolazione dei dati in tutte le maniere. E il divertimento viene limitato per lo più dalla larghezza di banda, ossia dalla mera questione di quanti dati si riescano a spostare a una data velocità".
Quali nuove tecnologie sono destinate ad alterare l'ambiente Amiga?
"CD-ROM, multimedia, drive ottici, tutti dispositivi che aiuteranno grafica e suono a migliorare, dando all'utente nel contempo tante possibilità in più. Più dati, capaci di spostarsi più velocemente...".
Giochi e informazioni, insomma, sono quasi la stessa musica...
"E' molto più ristretta la larghezza di banda che non la nostra immaginazione. Se la tecnologia continua a migliorare, non ci saranno limiti a ciò che l'immaginazione potrà produrre. Un giorno, potremmo perfino arrivare al gioco definitivo".

Cinemaware
La magia del cinema interattivo

Bob Jacob vuole realizzare film di cui siete i veri protagonisti, avventure che vi coinvolgono più di quelle reali.
"Il divertimento interattivo è un'esperienza, non solo un gioco" affer-

Bob Jacob

In ogni caso, Jacob sostiene che nei prossimi 12 mesi vedremo software per Amiga che cancellerà quello attualmente esistente, per un processo che lui vede come una selezione naturale. "Il Commodore 64 ha avuto bisogno di quattro o cinque anni prima che venissero sfruttate tutte le sue potenzialità: il 1990 e il 1991 saranno gli anni di Amiga".
Negli anni del decollo, dunque, cosa ci regaleranno gli sviluppatori in termini di software di intrattenimento? Qualcosa di nuovo, secondo Jacob. "Il vero divertimento è qualcosa di più di migliore grafica e suono più accurato. Con una macchina potente come questa, abbiamo bisogno di elaborare un nuovo concetto, più grande, di cosa dobbiamo intendere per software ricreativo".

Lui non ha dubbi sul suo obbiettivo: "Raccontare storie! Raccontare storie è il soggetto principale attorno a cui stiamo lavorando. Stiamo cercando la strada migliore di raccontare storie usando un personal computer. La nostra meta è sempre stata ricreare l'eccitazione che accompagna l'acquisto di un best seller. Vogliamo che l'utente sia coinvolto al punto di perdere il contatto con la realtà".
E qual è la cosa più difficile da fare? Jacob continua con il suo tono enfatico. "Sicuramente le interfacce. La migliore delle storie non potrà mai rimediare a un'interfaccia poco attraente. Il problema è che realizzare un'interfaccia semplice è più difficile che farne una complicata. E la vera semplicità richiede un sacco di programmazione difficilissima".

ma Jacob, fondatore e presidente della Cinemaware, produttrice di Rocket Ranger, TV Sports Football e altri programmi. E Amiga sarebbe la migliore piattaforma per queste esperienze? "Assolutamente! Amiga è il miglior personal computer sul mercato per grafica, suono, animazione, memoria e spazio su disco".
E cosa significa questa combinazione di doti, parlando di software di intrattenimento? "Amiga limita l'immaginazione molto meno di altri personal computer".
Quali sono le limitazioni di Amiga? Risponde Jacob: "I programmatori hanno una curva di apprendimento non ottimale. Amiga non è tanto un personal, quanto un minicomputer sotto mentite spoglie. Come risultato, molti programmatori si trovano in difficoltà quando cercano il sistema migliore per gestire la memoria della macchina".
Queste difficoltà hanno penalizzato lo sviluppo di software ricreativo su Amiga? Non più di tanto. Jacob ritiene che dipenda soprattutto dai programmatori, che devono imparare a sfruttare le risorse della macchina. "Negli ultimi tre anni abbiamo lavorato molto su Amiga, e solo da un anno o poco più padroneggiamo realmente il computer", dice.

I favoriti

E' ora di essere obbiettivi.
Abbiamo chiesto a una serie di personaggi autorevoli del mondo Amiga di indicare i loro giochi preferiti, tra quelli pubblicati da una società che non sia la loro.

Bob Jacob (Cinemaware) ha scelto StarGlider II della Rainbird:
"E' un'esperienza coinvolgente", spiega, "un mix riuscito di trama e stregoneria tecnologica".
R.J. Mical (Epyx) è uomo di molte parole, e molti giochi.
Dopo numerosi sforzi, siamo riusciti a limitare a quattro le sue scelte:
"Marble Madness, un gioco ipnotico. Flight Simulator II, il gusto di cambiare aeroplano a 10 mila metri di altitudine. Interceptor, un gioco che non ho ancora smesso di usare. E Sidewinder, superbo dal punto di vista audiovisuale".
A Rick Ross della Discovery piace quella che lui definisce "l'eleganza ipnotica di Shangai".
Stesso dicasi per Randy Linden della Visionary Design, che ammira l'estrema professionalità del pacchetto e spezza una lancia anche in favore di Sword of Sodan, che, a suo dire, "ha aperto nuove strade nel suo settore".
Joe Hubbard della Free Spirit va sui classici.
"Defender of the Crown combina grafica, trama e giocabilità. Non è difficile, ma è godibile e divertente".
Dice Eli Tomlinson della Scorpion:
"TV Sports Football è un grande mix di strategia e azione, con una grafica fantastica e un sacco di piccoli dettagli che abbelliscono il gioco".
Un altro fan di TV Sports Football è Manlio Allegra (IDG).
"Grande giocabilità, grafica eccellente e una superba interfaccia".
Il "nostro" Rhett Anderson ammira particolarmente Time Bandit per la sua varietà di situazioni:
"Non sai mai cosa potrà succedere il momento dopo".
Sempre per quanto ci riguarda, Keith Ferrell è innamorato di Reel Fish'n.
"Come nella vita reale, ci sono i giorni buoni e quelli meno buoni e da un momento all'altro ti metti a raccontare agli amici le dimensioni del pesce software che avevi quasi preso!".

R. J. Mical

Fatelo da soli

Siamo agli inizi di un'età dell'oro per il software di intrattenimento. Lo stato dell'industria è molto simile al mondo del cinema degli anni '20 a Hollywood. Come farne parte? Ecco una serie di consigli da persone stabilmente installate in questo business.
R.J. Mical: "Per scrivere un buon gioco, si deve capire che cosa vuol dire buon gioco. Studiare duramente. Imparare a programmare e a capire che cosa rende un gioco godibile". Giocare è importante.
"Bisogna giocare finché non cadono le orecchie, le dita non diventano insensibili e la vista non si annebbia. A quel punto, bisogna giocare ancora. E non solo videogiochi.
Giochi da tavolo, da scacchiera, di qualunque tipo.
Bisogna entrare, in un certo senso, nello spirito Zen del gioco, nella sua essenza. Solo allora si può creare un buon prodotto. A meno che, naturalmente, non abbiate una fortuna sfacciata. A un sacco di gente capita". Cercate incoraggiamento?
Ecco Bob Jacob: "Non è difficile, almeno negli Stati Uniti, entrare in questo mercato. Dipende dalle proprie ambizioni e aspirazioni. Il saper programmare e il saper creare un buon gioco sono doti concrete, dimostrabili, e c'è sempre spazio per persone desiderose di fare e imparare. Sappiate cogliere la vostra opportunità, e quando l'avete trovata lavorate. Abbiate tanta voglia di lavorare".
Eli Tomlinson della Scorpion dice: "Molta gente programma bene ma non riesce a esprimere le proprie idee. Prima di tutto deve venire una buona idea, e la programmazione viene dopo".
Bob Maludzinski, della Mindware International, tiene a sottolineare un punto importante:
"Cominciate con un sistema che vi costringa a saper programmare bene. Su un Commodore 64, per esempio, le poche risorse che avete vi obbligano a usarle nel modo migliore; non potete fare altro che imparare a dare il massimo".
Joe Hubbard, della Free Spirit, non rinuncia alla sua raccomandazione: "E' un grande momento per i programmatori di Amiga. Per cui, contattate un editore, ma solo quando sapete di essere diventati bravi: quello è il momento in cui fare vedere tutte le vostre capacità".
Prima di elargire consigli, Rick Ross della Discovery regala una battuta: "Qual è la differenza tra Dio e un programmatore? Dio sa di non essere un programmatore". Ma la risata non deve nascondere una realtà precisa.
"C'è sempre qualcuno migliore di voi. Molto poco di ciò che potete fare non è già stato fatto da qualcun altro. Per raggiungere il successo, il fattore numero uno è il saper lavorare in equipe. Il numero due?
Ricordarsi sempre che mettersi in affari è molto più complesso che saper programmare o scrivere un gioco".

Ma si può fare?
"Ogni giorno che passa, ci arriviamo sempre più vicini".

Epyx
Alla ricerca della sostanza

R.J. Mical sa di cosa parla, quando l'argomento è il software di intrattenimento. Mical ha sviluppato il Workbench e Intuition su Amiga originale, e ora si occupa di progetti speciali per Epyx, un colosso del gioco per computer. Ha opinioni ben precise su Amiga, e sui giochi da sviluppare per quest'ultimo.
"Nessuno può eguagliare la qualità degli effetti audio e video di Amiga", dice Mical. "Ma solo poche società hanno realmente sfruttato queste possibilità".
La situazione sta migliorando? "In generale, gli sviluppatori stanno producendo programmi più velocemente. Ma molti stanno convincendosi che Amiga è troppo complicato. Non sono ancora riusciti a capire quanto tempo gli ci vorrà per capire appieno il computer".
Come mai? "Amiga è più difficile da programmare degli altri computer. Chi lo ha fatto per anni ormai è un esperto, gli altri hanno bisogno di tempo. Ci sono molti Amiga nelle case, e quindi molti nuovi programmatori al lavoro".
Quali sono i problemi più difficili? Le stesse caratteristiche che fanno di Amiga un computer così eccitante. "Amiga ha le capacità e le funzioni proprie di computer molto più potenti, e saperle usare implica notevoli conoscenze".
Le differenze impongono diversi requisiti e la padronanza di nuove discipline. "Non è come sugli altri computer. Non ti puoi sedere davanti alla tastiera con il manuale rosso in mano (riferimento alla documentazione relativa all'Apple II. N.d.T.) e cominciare a programmare. La documentazione base di Amiga è alta una dozzina di centimetri, piena di complicazioni e dettagli. Il rischio di impiegare tempo senza riuscire a produrre è alto, e questo frena molti potenziali buoni programmatori".
"Ciò comunque non significa che i programmatori non stiano scrivendo giochi per Amiga. Significa solo che non stanno scrivendo buoni giochi. C'è in giro un sacco di gente, in aumento, che realizza prodotti senza conoscere la macchina". Il che si traduce in giochi poco spetta-

Manlio Allegra

Joe Hubbard

colari.
"Un grande gioco per Amiga deve essere perfetto" insiste Mical, rivelando il suo carattere da perfezionista. "Deve avvantaggiarsi di tutte le possibilità video e audio a disposizione". Ma c'è un altro ingrediente essenziale. "Deve essere un programma divertente e se non lo è, puoi scordartelo".
Il suono e la grafica possono perfino essere troppo seducenti. "Tantissimi giochi sono pieni di effetti veramente speciali ma risultano bruttissimi da giocare. Potrei farne una lista piuttosto lunga". Beh, non succede solo nel mondo dei giochi per computer. "Tutti effettacci e niente sostanza" dice Mical con il tono di uno che ha già visto tutto. "Ce ne sono in giro troppi".

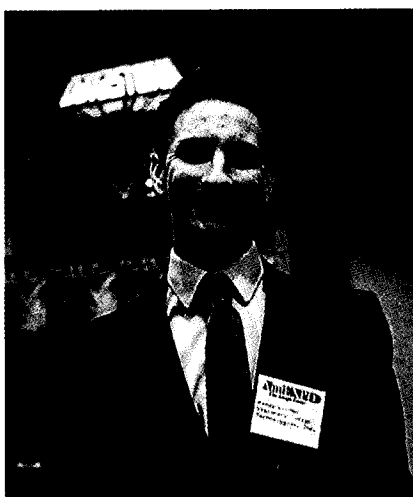
IDG
Il software SEGA

Manlio Allegra, uno dei partner dell'International Development Group (che "traduce" su floppy il software dei giochi da bar SEGA), cerca di vedere le cose sotto una prospettiva più ampia.
"E' importante che gli sviluppatori abbiano una visione globale del mercato" dice Allegra. "E' un mercato di dimensioni mondiali, e chi vuole avere successo dovrebbe considerarlo in questo senso".
Nel processo di conversione di un gioco il codice può viaggiare dal Giappone all'Europa al Canada o agli Stati Uniti. Ma molte cose rimangono sempre le stesse.
"La giocabilità innanzitutto" afferma Allegra, dichiarando di avere una notevole esperienza come videogiocatore. "Un grande gioco deve provocare un flusso di adrenalina". Per raggiungere lo scopo, i progettisti e i programmatori dello staff di Allegra non lesinano gli sforzi. "Azione, effetti sonori, musica, grafica: tutti gli elementi chiave per la riuscita di un gioco devono funzionare insieme per il massimo coinvolgimento del giocatore".
Allegra non fa nomi, ma non lesina nemmeno critiche a giochi per Amiga a suo avviso decisamente sopravvalutati. "Amiga è la migliore macchina sul mercato per quanto riguarda il suono", puntualizza, "ma queste capacità sono sfruttate da pochi. E poi tanti giochi vengono da trasposizioni di versioni scritte per Atari ST, una macchina inferiore".
Giocabilità, suono, grafica, azione; molte cose rimangono invariate nelle traduzioni da una lingua all'altra o da un sistema operativo all'altro. Ma quando si tratta di convertire un gioco da bar in una versione per Amiga, molte cose cambiano.
"Occasionalmente, dobbiamo alterare l'aspetto di alcuni personaggi, specialmente, strano ma vero, in molti giochi giapponesi, anche per il diverso gusto estetico del pubblico occidentale".
Quando tutti i dettagli sono a posto, però, l'adrenalina del giocatore fluisce davvero, e il linguaggio del gioco diventa quasi universale.
"Un grande gioco" dice convinto Allegra "non è diverso da un'opera d'arte".

Free Spirit
Divertimento per adulti

Joseph Hubbard, presidente della Free Spirit, sente arrivato il momento di proporre software di intrattenimento più indirizzato agli adulti. Non pornografia, chiaro, ma vero divertimento, con in più un pizzico di erotismo. Il suo grande successo, Sex Vixens From Outer Space (qualcosa tipo "Bisbetiche del sesso dallo spazio profondo". N.d.T.) chiarisce subito il suo tipo di approccio.
Che sia per adulti o per bambini, comunque, i requisiti per il software di intrattenimento su Amiga sono rigidi. "Buona grafica, effetti sonori, buona trama, ma specialmente la grafica è importante. Amiga è noto soprattutto per questo, no?"
E' più importante avere sottomano un buon gioco, o un buon gioco per Amiga? Hubbard non vede molta differenza. "Se vuoi creare un grande gioco, puoi creare un grande gioco, e implementarlo su Amiga". Questo significa avvantaggiarsi delle caratteristiche della macchina. E non tutti lo fanno. "Alcuni giochi per Amiga puntano solo sulla grafica e nient'altro" puntualizza Hubbard con evidente disapprovazione. Potrebbe essere che sia Amiga a tentare i programmatori sotto questo aspetto, viste le sue doti. "Ovviamente" precisa con un sorriso "è difficile che si raggiunga il successo con un gioco incentrato sulla grafica in ambiente IBM".
Soprattutto dopo avere raggiunto il successo con Vixens, Hubbard è consapevole dei dubbi che lo stile dei suoi giochi possono provocare negli ambienti più benpensanti della società. In che punto è situata la

linea di demarcazione tra erotismo e pornografia, tra divertimento e vizio? Hubbard ne fa una questione strettamente personale.
"Quando abbiamo realizzato Sex Vixens, pensavamo a un vecchio film soft-core dal titolo "Flesh Gordon" (gioco di parole tra Flash Gordon, noto personaggio della fantascienza degli anni '60, e "flesh", carne. N.d.T.). Pensavamo di dare al pubblico un po' di erotismo e tanto divertimento, non necessariamente in quest'ordine". Come risponde Hubbard alle accuse secondo cui il software a contenuto erotico è più pericoloso di altre forme similari di intrattenimento, a causa dell'interattività propria del computer? Per lui, si tratta di opinioni fuorvianti.
"Tutti hanno qualche fantasia, e non c'è niente di sbagliato in questo. Un computer non può fare più danni di quelli che l'individuo può già farsi da solo".
E si infervora: "Preferisco vedere la gente giocare con software erotico piuttosto che con giochi violenti. Chiunque può sedersi al computer e sterminare centinaia di alieni, o di esseri umani con una uniforme da nemico, per non dire d'altro. Pensate che questo sia più salutare o meno pericoloso del sesso?" ▲

Randy Linden

Divisi dall'Oceano

Eli Tomlinson

Antony Jacobson

Più di 200 anni dopo la rivoluzione americana, ci sono ancora differenze tra i compassati britannici e i più informali statunitensi. Due esperti guardano alle differenze tra il software ricreativo inglese e quello americano.

Eli Tomlinson
L'aquila americana

Eli Tomlinson, presidente della Scorpion Software, importa una certa quantità di software inglese negli USA. A suo parere, le differenze non sono tanto di ordine qualitativo quanto di gusti e tendenze.
"La qualità della programmazione è equivalente", dice Tomlinson, "non per niente i programmatori inglesi hanno una certa fama".
Tale qualità porta però a differenti risultati. "I nostri gusti e i loro tendono a differenziarsi molto. Gli americani sono attratti dai giochi di riflessione, di ruolo, di genere fantasy, dai wargame" aggiunge Tomlinson.
E in Inghilterra? "Gli inglesi preferiscono i classici giochi da bar, gli shoot'em-up". Ci sono quindi differenti approcci al mercato. "Gli Stati Uniti sono un mercato molto più chiuso dell'Inghilterra. Là, chiunque può cominciare a produrre giochi e piazzarli in tutti i punti vendita del Regno Unito". Per Tomlinson, in America questo è più difficile. "In USA, bisogna trattare con un certo numero di distributori, alcuni dei quali non mettono a listino più di una certa quantità di prodotti. Qualche volta, la minima possibile. Naturalmente, le case più forti hanno i posti migliori sugli scaffali. Così diventa complicato arrivare ai punti vendita". In Inghilterra, invece, lo spazio sugli scaffali è garantito, ma la vita di un prodotto è breve."Non più di due settimane: o si dispone di un vero supergioco, o non si passa la fase iniziale degli ordini".

Antony Jacobson
Al servizio della Corona

Antony Jacobson, editore, è tanto inglese quanto appassionato di Amiga. Questa combinazione di tratti dà origine ad alcune opinioni precise sui lati buoni del software europeo e sulle caratteristiche di quello americano. Sono simili? Jacobson sostiene di no. "C'è una differenza sostanziale: i giochi statunitensi sono più semplici. Buoni, veloci, ma sono in fondo shoot'em-up. In Europa è diverso. Si producono giochi più complessi, come Captain Blood e Starglider. Questi giochi sono piuttosto complessi, e vengono richiesti da un pubblico in possesso di una notevole esperienza in fatto di gioco, che vuole software d'avanguardia. In questo momento non è possibile produrre un gioco per Amiga sul mercato e venderlo, se non è un gioco di cui c'è grande richiesta".
Quali sono i migliori giochi americani? "Mi piacciono Sword of Sodan e Arkanoid". Jacobson è fiducioso in un affiorare di nuove energie creative nel mercato a stelle e strisce.
"Ci sono programmatori sempre migliori, società alla ricerca di innovazioni interessanti. Secondo me i migliori in questo campo sono quelli della Cinemaware. Loro stanno veramente facendo scuola".

Visionary Design Technologies
Sudore e dettagli

Randy Linden pensa che siano le piccole cose a fare la differenza tra un gioco e un grande gioco.
Direttore dello sviluppo dei prodotti per la Visionary Design Technologies, editrice di Vortex, Linden sente che la fase finale della realizzazione, quella dedicata ai dettagli, sia la più cruciale nella creazione di un gioco.
Certo, il quadro di insieme è importante. I requisiti per un gioco di successo su Amiga? "Animazione di qualità, grafica di alta qualità, e suono pulito, di altissima qualità" risponde senza esitazioni Linden; poi si ferma un momento, riflettendo sulla questione del suono e dello scarso successo ottenuto da alcuni produttori nel cercare di sfruttarlo al massimo.
"Un sacco di suoni su Amiga sembrano banalmente digitalizzati, anziché prodotti con professionalità". "Questo è sbagliato", afferma Linden, "soprattutto perché è evitabile. Amiga ha un potenziale di suono e grafica sconosciuto agli altri computer. Invece, molte società si accontentano di pubblicare prodotti in serie, senza curarli nei dettagli come potrebbero".
I giochi per Amiga sono intrinsecamente differenti da quelli per altri computer? "Credo di sì. Poiché Amiga è molto veloce, ci sono più giochi che puntano a stimolare i riflessi. Per esempio, la maggior parte degli ultimi giochi hanno personaggi coloratissimi e molta azione, in maniera molto simile ai giochi da bar; in questo senso, Amiga è quasi una macchina da bar, e in ciò è molto diverso da IBM, quindi dai giochi che su quest'ultimo vengono creati. Linden è sicuro che stia per cominciare un'età dell'oro per i programmatori Amiga. I migliori sviluppatori stanno cominciando a sfruttare al meglio il computer. "Giochi come Dragon's Lair e Sword of Sodan stanno portando il livello dei giochi per Amiga più in alto. Verso un livello nuovo e più avanzato".
Questo nuovo livello farà ricorso a nuove tecnologie? CD-ROM, per esempio?
"Sfortunatamente", dice Linden, "ci sono due tipi di utenti Amiga: chi ha un 500 e chi ha un 2000. Il CD-ROM offre possibilità eccezionali, ma penso che potranno beneficiarne solo i possessori di un 2000".
Comunque si sviluppi la tecnologia, Linden prevede un miglioramento del livello del software di intrattenimento su Amiga. "Le prossime ondate di giochi saranno estremamente creative, basate su nuove idee. I nuovi giochi saranno come libri, dotati di eccezionale profondità e grandi possibilità di coinvolgimento e di immedesimazione".
E poi? "La gente vuole qualcosa in più, sempre. A nessuno basta sedersi al computer e cominciare a sparare. La gente vuole giochi che la intrattenga, nel senso più esteso della parola. Giochi che facciano pensare, imparare, scoprire cose nuove". ▲

Le Finestre

Gianni Biagini

Forse le avrebbero potute chiamare aree applicative, però converrete anche voi che il termine finestre è molto più poetico e dà l'idea dell'interfaccia utente che guarda verso il mondo, verso la libertà.
Tralasciando le digressioni semantiche, avviciniamoci al soggetto di questo mese: le finestre.
Tutti noi abbiamo presente la gestione dello schermo di lavoro di Amiga, il WorkBench per intenderci; esso è composto da uno SCREEN (uno sfondo con alcune particolarità che vedremo), da una serie di ICONE (quelle simpatiche figurine che rappresentano i programmi e i dischi) e dalle finestre vere e proprie.
Accendiamo Amiga e inseriamo il WorkBench; dopo alcuni secondi si presenta una videata con l'immagine di un dischetto e cliccando due volte sulla stessa vediamo finalmente una prima finestra. Essa rappresenta il contenuto del disco, ha come titolo infatti WorkBench e se clicchiamo due volte sul cassettino System si apre un'altra finestra che ne rappresenta il contenuto. Abbiamo quindi una prima idea delle potenzialità del sistema a finestre e possiamo senza utilizzare comandi come CD per cambiare directory, fare riferimento a un particolare programma, semplificando al massimo i comandi da utilizzare.
Oltre che a rappresentare degnamente le directory, le finestre possono contenere delle immagini grafiche, semplificare meccanismi di input/output e chi più ne ha più ne metta.
Nasce quindi l'esigenza di capire a fondo le problematiche legate all'utilizzo di questo geniale sistema di comunicazione con la macchina. In questa sede vedremo come inizializzare finestre, come operare con esse e quali sono gli attributi fondamentali applicabili alle stesse.
Premettiamo che la trattazione si dividerà in due parti distinte: la prima che analizza le problematiche legate ad un linguaggio di lavoro quale il C (i riferimenti saranno compatibili con i due standard Lattice ed Atzec) mentre la seconda ricoprirà le più semplici modalità operative messe a disposizione dal linguaggio Basic.
Iniziamo dal linguaggio C: le finestre sono viste come delle strutture dinamiche costituite da una serie di campi da specificarsi in fase di creazione della finestra stessa; ma andiamo per gradi e vediamo cosa è approssimativamente una struttura. Per chi conosce il Pascal basterà dire che è un record, per gli altri è una serie di dati che fanno riferimento a un soggetto specifico, come si vede in tabella 1.
Questi appena mostrati sono due record che associano ad ogni anagrafica (il nome), 3 campi diversi: leggendo la struttura, so che il Sig.Rossi di 45 anni ha il numero telefonico 02/34567 e che mi ha pagato una somma di 120000 lire. Lo stesso posso fare leggendo i dati relativi al signor Bianchi, poiché ad ogni nominativo sono associate delle informazioni.
Un record è quindi una struttura composta da campi (in questo caso sono nome, età, telefono, e saldo); è bene sapere che nei linguaggi di alto livello il record è da considerarsi una tra le strutture più importanti poiché permette di creare dati che hanno una grossa flessibilità e semplicità di uso.
Torniamo al buon linguaggio C: i record ovviamente esistono e si chiamano strutture (STRUCT), anche le finestre, come vedremo tra poco, vanno gestite come strutture e hanno la seguente forma:

```
struct NuovaFinestra
  {
   SHORT   Angolo_Sinistro,
Angolo_Superiore,
         Larghezza,Altezza;
    UBYTE    Penna_Del_Contorno,
Penna_Dell_Interno;
   USHORT IDCMPFlags;
   ULONG  Flags ;
   struct Gadget *Primo_Gadget;
   struct Image  *CheckMark;
   UBYTE  *Titolo;
   struct Screen *Screen;
   struct BitMap *BitMap;
    SHORT      Larghezza_Minima,
Altezza_Minima,

Larghezza_Massima,Altezza_Massima;
   USHORT Tipo;
  };
```

NOME	età	telefono	saldo
Sig. ROSSI	45	02/34567	1200000
Sig BIANCHI	48	02/45678	2300000

Tab. 1

Cosa vorrà dire tutto ciò che abbiamo scritto? Abbiamo detto che una finestra è una struttura, e l'abbiamo definita.
Vediamo ora cosa sono quelle strane componenti tra le parentesi graffe: senza entrare nel merito dei tipi di variabili che esistono in C (è un buon riferimento "Il linguaggio C" scritto da Kernigan e Ritchie, i creatori del linguaggio stesso ed edito dal Gruppo Editoriale Jackson) possiamo esaminare uno per uno i campi del record NuovaFinestra. Quando possibile i nomi dei campi sono stati inseriti in italiano per facilitare la comprensione e rendere meno pesante la lettura; i primi quattro campi sono di tipo SHORT (mi raccomando di scriverlo in maiuscolo e di iniziare il programma con l'istruzione #include "exec/types.h" che dice al programma che vi sono altri tipi di variabili oltre a quelle standard di cui SHORT non fa parte); essi rappresentano la posizione e l'estensione della finestra. Seguono due campi di tipo UBYTE (anche questo in maiuscolo) che dicono al sistema operativo di usare, nel disegnare le finestre, i due colori dal numero specificato.
Arriviamo ora ad un campo particolarmente importante che analizzeremo meglio in seguito: gli IDCMP-Flags (altro non è se non l'abbreviazione di Intuition Direct Communication Message Port Flags) che sono delle particolari flag che determinano le modalità di Input/Output delle finestre.
Seguono i flag del tipo della finestra (anche questi verranno analizzati in seguito) e il puntatore ai gadget non standard (supponiamo di voler creare un gadget che espanda la finestra alla grandezza dello schermo: dovremo inizializzare una struttura di gadget e connetterla alla finestra passando in questo campo il puntatore al gadget stesso).
Da notare che pur avendo più gadget non standard, si deve comunque passare il puntatore al primo poiché gli altri sono collegati fra loro attraverso una struttura a lista.
Tralasciamo il campo CheckMark per semplicità e vediamo il seguente, quello che permette di dare un nome alla finestra.
Finalmente la nostra creazione esce dall'anonimato e grida al mondo la sua identità, non più finestre sconosciute.
Se dare un nome ad una finestra può sembrare inutile, pensate a WorkBench, l'interfaccia utente del nostro Amiga: ogni directory è rappresentata da una finestra e visto che è possibile fare riferimento a più directory contemporaneamente, assegnare il nome ad ognuna delle window che le rappresentano comporta una migliore intelliggibilità del contenuto, per esempio, del disco.
Finiamo di analizzare i campi della struttura: incontriamo il puntatore ad uno schermo, lì metteremo l'indirizzo della struttura di screen da noi definita (se usiamo lo schermo del WorkBench assoceremo al campo il valore NULL ossia puntatore al nulla). Sempre per semplicità lasciamo i puntatori ad un BitMap, ossia disegni che vogliamo che compaiano nella finestra, e vediamo il gruppo di variabili SHORT che segue: Larghezza_ Minima, Altezza_Minima, Larghezza_Massima, Altezza_ Massima rappresentano rispettivamente ognuna delle minime e massime dimensioni che può avere la finestra.
Ultimo punto da prendere in considerazione è il tipo della finestra che stiamo definendo: esistono due soli valori che possono essere inseriti in questo campo, essi sono WBENCH-SCREEN e CUSTOMSCREEN che dicono alla struttura se si vuole usare come screen quello del WorkBench o uno definito dall'utente (in questo caso si deve mettere il puntatore alla struttura di screen nel campo NuovaFinestra.screen).
Possiamo ora iniziare a tirare le somme: abbiamo una struttura definita nel modo visto e vorremmo quindi vedere una finestra sullo schermo. Come possiamo fare?
Abbiamo detto che affinché Amiga possa conoscere alcuni particolari tipi di variabili, dobbiamo connettere al programma la libreria TYPES.H; non ci resta che farlo come prima istruzione:

```
#include "exec/types.h"
```

Dove trova Amiga le specifiche delle strutture usate da Intuition (e quindi la struttura associata ad una finestra)?
Naturalmente in una libreria che si chiama Intuition.H:

```
#include "intuition/intuition.h"
```

Adesso Amiga è stato istruito: ora sa come gestire le finestre, non ci servono altre librerie e possiamo partire con il programma vero e proprio (le precedenti istruzioni erano precedute dal segno # poiché erano direttive per il precompilatore).

Abbiamo le strutture, usiamole!

```
struct IntutionBase *IntuitionBa-
se;
struct Window *Window;
```

cosa abbiamo fatto? Niente di arcano, abbiamo solo detto alla macchina che le variabili IntuitionBase e Window non sono altro che puntatori a strutture di tipo IntuitionBase e Window.
Non abbiamo ancora detto che per aprire una libreria è necessario consigliare alla macchina quale versione usare (si potrebbe anche fargli scegliere la prima che gli capita a portata di mano, solo che per ragioni di compatibilità noi specifichiamo che vogliamo la versione 29).

```
#define INTUITION_REV 29
```

con l'ultima istruzione diciamo al precompilatore (sempre quello di prima) che vogliamo che INTUITION_REV sia una costante e che abbia valore 29.
Ora siamo finalmente pronti, abbiamo definito le strutture che ci servivano, il nostro programma continua con:

```
main()
{
   struct NewWindow NewWindow;
```

```
  int n,m;

   IntuitionBase=(struct  Intui-
tionBase *)
   OpenLibrary("intuition.library",
INTUITION_REV);

  NewWindow.LeftEdge=30;
  NewWindow.TopEdge=30;
  NewWindow.Width=100;
  NewWindow.Height=100;
  NewWindow.DetailPen=0;
  NewWindow.BlockPen=1;
  NewWindow.Title="AmigaMagazine";
  NewWindow.Flags=ACTIVATE|SMART_REFRESH;
  NewWindow.IDCMPFlags=NULL;
  NewWindow.Type=WBENCHSCREEN;
  NewWindow.FirstGadget=NULL;
  NewWindow.CheckMark=NULL;
  NewWindow.Screen=NULL;
  NewWindow.BitMap=NULL;
  NewWindow.MinWidth=20;
  NewWindow.MinHeight=20;
  NewWindow.MaxWidth=300;
  NewWindow.MaxHeight=150;

/* con le precedenti istruzioni
specifichiamo i parametri  carat-
teristici della finestra */

 Window=(struct Window *) OpenWin-
dow (&NewWindow);

  for (n=1;n<=10000;n++)
    for(m=1;m<=1000;m++);

/* le precedenti istruzioni servo-
no solo per perdere tempo */

  CloseWindow(Window);

/* Chiudiamo la finestra prima di
uscire */

}
```

Abbiamo creato la finestra, l'abbiamo fatta apparire e dopo un po' di tempo essa è sparita. Benissimo, a questo punto non ci resta che esaminare i flags per avere una trattazione sufficientemente completa delle finestre. Per quel che riguarda gli IDCMPFlags, li prenderemo in considerazione solo dopo aver abbondantemente parlato dell'Input/ Output secondo Intuition.
Nella struttura della finestra avevamo incontrato un campo detto Window.Flags; i valori inseribili sono 10, eccoveli in ordine rigorosamente alfabetico:

ACTIVATE attiva la finestra appena essa viene aperta.

ACTIVEWINDOW mette il programma al corrente che la finestra a lui associata è attiva.

BACKDROP crea una finestra che sta sotto a tutte le altre.

BORDERLESS la finestra creata risulta sprovvista del caratteristico rettangolo che ne delimita l'estensione.

INACTIVEWINDOW come ACTIVEWINDOW solo che avverte se la finestra è inattiva.

NOCAREREFRESH indica che la finestra non deve essere rinfrescata dopo ogni cambio.

REPORTMOUSE aziona il meccanismo che intercetta i movimenti del mouse.

SIMPLE_REFRESH specifica che il ridisegno del contenuto della finestra è ad opera del programma.

SMART_REFRESH il ridisegno è fatto automaticamente.

SUPER_BITMAP specifica che la finestra deve essere disegnata mediante una BITMAP (maschera di bit) definita dall'utente.

Sul disco è presente una procedura simile a quella descritta anche lei semplificata al massimo affinché si possano notare subito le particolarità di cui ci occupiamo.
Trovate che quanto detto sia difficile e non sufficientemente gratificante? Certo il linguaggio C è complesso come utilizzo, però dà delle potenzialità che altri linguaggi difficilmente riescono a mettere a disposizione: è il caso del Basic dove per fare una finestra basta una linea di programma, ma lavorare per esempio usando le funzioni di sistema operativo più nascoste è particolarmente complicato.

Il linguaggio Basic mette a disposizione dell'utente il comando WINDOW (guarda caso) che accetta come parametri il numero della finestra seguito da (sono tutti parametri opzionali) titolo, rettangolo di estensione, tipo e identificatore di schermo.

Il comando:

```
WINDOW   1,"Amiga   Magazi-
ne",(10,10)-(100,100),15
```

apre una finestra assegnandole il numero 1, il titolo "Amiga Magazine" le coordinate 10,10 e l'estensione di 90 punti per 90 (100,100 è infatti l'angolo inferiore sinistro).
Resta infine il tipo della finestra e guardiamo, per capire meglio, la tabella seguente:

1• la finestra può essere ridimensionata attraverso l'apposito gadget
2• la finestra può essere spostata
4• la finestra può essere messa in primo o in ultimo piano attraverso l'apposito gadget
8• la finestra può essere chiusa attraverso l'apposito gadget in alto a sinistra
16•indica ad Amiga di tenere memoria del contenuto della finestra affinché se prima viene coperta e poi rimessa in vista il suo contenuto non si cancelli.

Nell'esempio precedente abbiamo specificato tipo 15 ossia la somma di 1 (ridimensionamento) più 2 (riposizionamento) più 4 (profondità) più 8 (chiusura), infatti 1+2+4+8=15 per cui la finestra comparirà con i quattro gadget che corrispondono alle funzioni specificate sopra.

Per quel che riguarda l'identificatore di schermo, si deve mettere il numero assegnato ad uno schermo precedentemente creato; se il campo viene omesso il Basic accetta -1 e usa per lo schermo quello del WorkBench. ▲

Acquisition V.1.3F

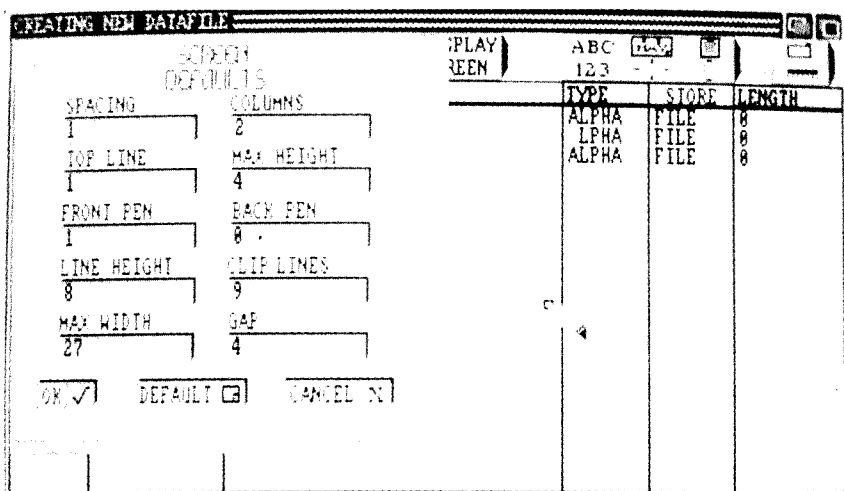
Finalmente anche in Italia l'ultimo aggiornamento del famosissimo programma Acquisition !

Marco Tortolina

I numerosi programmi DataBase (DB) rivolti agli utenti di Amiga, sottoposti a continui e sofisticati aggiornamenti, non si limitano ormai più al ruolo di semplici agende elettroniche sulle quali annotare indirizzi e nomi di amici e parenti. Infatti i migliori DB "relazionali" oggi in circolazione (Superbase professional, Dataretrieve, etc.), hanno caratteristiche tali per cui è possibile la gestione di archivi di testo, suoni e immagini oltre alla possibilità di intervenire personalmente nell' inserimento, ricerca e relazione dei dati grazie ad un proprio linguaggio di programmazione. Tutto questo è possibile anche con ACQUISITION, il DB prodotto dalla TAURUS IMPEX, di cui prenderemo in considerazione la versione 1.3F, l'ultimo aggiornamento di un pacchetto commercializzato all'inizio del 1987 (release 1.2) dopo essere stato in lavorazione per più di tre anni. In ACQUISITION le prerogative di un buon DB sono molto ben sviluppate; si pensi alla capacità di amministrare contemporaneamente più archivi in relazione tra loro, ognuno contenente fino a 100 milioni di record, suddivisi a loro volta in 10 milioni di campi, lunghi 10 Mbyte! Il potente linguaggio interno, chiamato ACOM, è poi il vero fiore all'occhiello: si compone di circa 200 comandi simili al BASIC, quindi facile da usare anche per il neofita, che permettono un totale controllo del programma.
Siamo senza dubbio di fronte ad un software complesso e articolato che sembra non risentire di qualche annetto di anzianità ma che pare invece migliorare di versione in ver-

sione rimanendo al passo con DB più recenti e innovativi. Vediamo ora come è organizzato ACQUISITION. Il programma è su due dischi: MAKE che, come prenderemo in esame seguentemente, crea ed imposta gli archivi sui quali riverseremo i dati e FILE con il quale ci sarà consentito manipolarli. Nella confezione è presente il classico disco di esempi abbastanza chiaro e di utile consultazione. Il tutto è completato da un manuale, purtroppo in inglese, che per quanto esauriente e comprensibile non mancherà certo di incutere il timore (sono 300 pagine!) di non potercela mai fare ad usare ACQUISITION in maniera completa. Naturalmente i dischi non sono protetti per avere la possibilità di lavorare con copie di backup anche se per far girare il programma è necessario installare il "dongle" in dotazione. La configurazione di sistema prevede almeno due disk drive, ma, vista la struttura del software, un hard disk sarebbe caldamente consigliato. Utile potrà essere una eventuale espansione di memoria, ormai quasi indispensabile per gli utenti più esigenti di Amiga.

Struttura di Acquisition 1.3F

MAKE è il disco che useremo per dar forma alle nostre idee e su come vorremo organizzare il DB. Il doppio Click sull'icona di CREATING, il primo programma da caricare, ci porterà in un perfetto ambiente Amiga, con mouse, gadget e menu a discesa. Questo modulo definisce, suddivisi in campi (fields), il numero di dati e il tipo di informazioni in essi contenuti, scelti tra: alfanumerici, numerici, data, ora, immagini e suoni IFF e altri definiti in un formato personale. E' possibile assegnare ad ogni campo una sequenza (stream) di comandi ACOM: questi miniprogrammi permettono di ottenere delle convalide, inserire operazioni o assegnare un qualsiasi simbolo da noi desiderato. Comodo sarà utilizzare variabili temporanee in memoria. CREATING si chiude mostrando un casellario su cui sarà scelto un indice (index) in base al quale verranno memorizzati in ordine i dati su un disco vuoto che avremo in precedenza formattato grazie all'utility INSTALL (da non confondere con il comando install usato in CLI), presente sia in MAKE che in FILE. Per usare ACQUISITION pienamente come DB "relazionale" dobbiamo stabilire dei percorsi tra i files; BRIDGING ci permette di costruire un "path" per collegarli, l'operazione è più facile da eseguire che da spiegare, basterà seguire attentamente le indicazioni sulla schermata, scegliere gli archivi che si vorranno congiungere e i campi che saranno relazionati in fase di inserimento dati, inoltre, con BRIDGING si potranno selezionare indici supplementari. Mentre i due precedenti programmi si limitavano a definire dei progetti, con PASTING possiamo ora manipolarli per poter dar loro la forma di inserimento e uscita dati da noi voluta. E' possibile abbellire lo schermo con bordi, linee e testo; ci sarà, inoltre, consentito cambiare posizione ai campi ridimensionandoli a nostro piacere, il tutto per avere una migliore comprensione dell'archivio creato. Il compito di MAKE è così terminato. Ora che tutto è impostato e pronto a ricevere i dati passiamo al contenuto del disco FILE. FILING è il quarto programma di cui è composto ACQUISITION ed è qui che vengono inseriti i dati. Esso può manipolare un project per volta il quale è in grado di contenere fino a 16 database. Vari comandi, di più frequente utilizzo, sono attivabili tramite una serie di icone oltre che da menu. Tra questi segnaliamo il potente "search" con

cui possiamo trovare, aggiornare, salvare o cancellare i records che corrispondono ad un particolare criterio. Con CALCULATOR poi siamo in grado di effettuare operazioni con i dati contenuti nei campi. Abbiamo oltretutto l'opzione di far "parlare" le stringhe unite a campi e icone, di mostrare disegni ed eseguire musiche definite nei record. Da notare che FILING immagazzina i record in memoria per cui se da una parte ne deriva una grande velocità di gestione, dall'altra, un reset non voluto o una interruzione di elettricità causerebbero la perdita irrimediabile dei dati immessi. Con REPORTING viene affrontato l' importante aspetto di estrazione e presentazione delle informazioni contenute nei vari archivi. Il modulo, tramite numerose funzioni, ci produrrà la ricerca desiderata indirizzandola verso lo schermo, la stampante o un altro dispositivo. REPORTING può essere designato per presentare dati in molti formati differenti, ad esempio, inserire nomi e indirizzi ricavati da record, in una lettera standard (mailmerge); presentare tabelle in forma colonnare con parziali e totali; stampare fogli ad uso fatture, contratti ecc.
Completa ACQUISITION il programma SETTINGS che può essere usato per settare varie opzioni come ad esempio un particolare device d' uscita o dotare i programmi di uno schermo proprio. Qui troviamo anche l' opzione di import-export che consente il trasferimento da e verso files di testo. Se poi si volesse italianizzare almeno qualcosa, potremo cambiare nome ai comandi ACOM.

Conclusioni

Senza un hard disk, le grandi capacità di ACQUISITION 1.3F si pagano purtroppo in termini di tempo: si pensi al fatto di dover passare da sei distinti programmi e inoltre, a causa della complessità di questi, si dovranno necessariamente cercare istruzioni sul manuale che come già detto è in inglese. SUPERBASE PROFESSIONAL, il concorrente diretto di ACQUISITION, non ha la stessa estendibilità di campi (ma a chi serve un campo di 10 milioni di caratteri?) ed alcune utilità come il calcolatore o il riordino dati programmabile, ma è di più semplice utilizzo. Senza dubbio altri DB sono meno potenti ma anche più comodi e per funzionare richiedono solo la configurazione base di Amiga. ACQUISITION è quindi più adatto in ambiente professionale dove può esprimere in pieno le sue doti, considerando che solo con un uso continuo e interessato si arriva a comprendere l'intera potenzialità del programma. ▲

Acquisition è distribuito da: C.T.O. s.r.l - Via Piemonte 7/F- Zola Predosa (BO) - Tel.051/753133

I Servizi di AMIGA MAGAZINE

Amiga Magazine offre una serie di servizi per agevolare i propri lettori nel reperimento di software e materiale utile alla programmazione. E' disponibile l'intera libreria di dischetti di pubblico dominio curata da Fred Fish. Ogni dischetto contiene numerosi programmi e utility, spesso corredati da listati sorgenti e commenti degli autori.
Per districarsi fra le centinaia di programmi disponibili nei dischi di Fred Fish, come in precedenza su Transactor per Amiga, pubblicheremo nei prossimi mesi l'elenco ragionato.
Tale elenco riporta, divisi per categoria, tutti i programmi presenti, completandoli con informazioni quali la descrizione della funzione, l'autore, il numero di versione, la disponibilità del sorgente e il disco nel quale sono contenuti. I dischetti possono essere ordinati contrassegnando i numeri desiderati, purché la quantità sia di almeno 3 dischi.

BUONO D'ORDINE

Completare il buono d'ordine (o una sua fotocopia) e spedire in busta chiusa a:
I servizi di Amiga Magazine, Via Pola 9 - 20124 Milano

Si può allegare: assegno, contanti o fotocopia della ricevuta di versamento **c/c n.11666203** intestato a **Gruppo Editoriale Jackson.** Non si effettuano spedizioni in contrassegno.

Desidero ricevere i seguenti articoli; contrassegnare con una X i numeri di Fish Disk desiderati (minimo 3)

❑ 1	❑ 12	❑ 23	❑ 34	❑ 45	❑ 56	❑ 67	❑ 78	❑ 89	❑ 100	❑ 111	❑ 122	❑ 133	❑ 144	❑ 155	❑ 167
❑ 2	❑ 13	❑ 24	❑ 35	❑ 46	❑ 57	❑ 68	❑ 79	❑ 90	❑ 101	❑ 112	❑ 123	❑ 134	❑ 145	❑ 156	❑ 168
❑ 3	❑ 14	❑ 25	❑ 36	❑ 47	❑ 58	❑ 69	❑ 80	❑ 91	❑ 102	❑ 113	❑ 124	❑ 135	❑ 146	❑ 157	❑ 169
❑ 4	❑ 15	❑ 26	❑ 37	❑ 48	❑ 59	❑ 70	❑ 81	❑ 92	❑ 103	❑ 114	❑ 125	❑ 136	❑ 147	❑ 158	❑ 170
❑ 5	❑ 16	❑ 27	❑ 38	❑ 49	❑ 60	❑ 71	❑ 82	❑ 93	❑ 104	❑ 115	❑ 126	❑ 137	❑ 148	❑ 159	❑ 171
❑ 6	❑ 17	❑ 28	❑ 39	❑ 50	❑ 61	❑ 72	❑ 83	❑ 94	❑ 105	❑ 116	❑ 127	❑ 138	❑ 149	❑ 160	❑ 172
❑ 7	❑ 18	❑ 29	❑ 40	❑ 51	❑ 62	❑ 73	❑ 84	❑ 95	❑ 106	❑ 117	❑ 128	❑ 139	❑ 150	❑ 161	
❑ 8	❑ 19	❑ 30	❑ 41	❑ 52	❑ 63	❑ 74	❑ 85	❑ 96	❑ 107	❑ 118	❑ 129	❑ 140	❑ 151	❑ 162	
❑ 9	❑ 20	❑ 31	❑ 42	❑ 53	❑ 64	❑ 75	❑ 86	❑ 97	❑ 108	❑ 119	❑ 130	❑ 141	❑ 152	❑ 163	
❑ 10	❑ 21	❑ 32	❑ 43	❑ 54	❑ 65	❑ 76	❑ 87	❑ 98	❑ 109	❑ 120	❑ 131	❑ 142	❑ 153	❑ 165	
❑ 11	❑ 22	❑ 33	❑ 44	❑ 55	❑ 66	❑ 77	❑ 88	❑ 99	❑ 110	❑ 121	❑ 132	❑ 143	❑ 154	❑ 166	

Nota: il n° 164 non è disponibile

❑ 3 Fish Disk L. 24.000 (abbonati L.18.000)

❑ L. 5.000 per ogni disco aggiuntivo riferito ad ogni singolo ordine

❑ Interfaccia MIDI L. 92.000

Tutti i prezzi sono da intendersi IVA inclusa e spese di spedizione comprese.

Firma ______________________

(se minorenne quella di un genitore) Gli ordini non firmati non verranno evasi.

Cognome ______________________

Nome ______________________

Via ______________________

Cap __________ Città ______________

Prov __________ Telefono ______________

Codice abbonamento ______________

MIDI Computer e musica

Un super libro per conoscere tutti i segreti del misterioso mondo MIDI

Claudia Sforzini

Nel nostro tempo, il settore a cui è rivolto questo libro, deve necessariamente mediare una cultura musicale ad un nozionismo, in alcuni casi esasperato, per il corretto impiego delle nuove tecnologie, associate all'utilizzo delle nuove apparecchiature elettroniche musicali e, nel contempo, evitare che venga sopraffatta l'attività creativa, che deve comunque essere il punto fermo di riferimento dell'artista.

Il musicista che intende servirsi delle nuove tecnologie (ed è difficile supporre che ne possa fare a meno) deve quindi assumere le conoscenze necessarie per operare correttamente sulle attrezzature elettroniche, traendo da esse il massimo profitto possibile. Il MIDI, acronimo di Musical Instrument Digital Interface (che significa Interfaccia Digitale per Strumenti Musicali), è oggi una realtà universalmente accettata e le interfacce MIDI sono di fatto incorporate in tutti gli strumenti elettronici. Il sempre crescente successo del MIDI ha sviluppato nel tempo la progettazione di hardware e software dedicati, dando un impulso determinante alla costituzione di una nuova disciplina specializzata: l'informatica musicale.

L'ausilio di questi nuovi media ha consentito al musicista un controllo, in termini di qualità e quantità, delle proprie apparecchiature, prima impossibile, espandendone enormemente le possibilità sonore. Il libro vuole quindi essere da tramite, o se vogliamo una sorta di interfaccia, comunque a senso unico, pensata per il musicista che vuole essere al passo con il suo tempo, non rinunciando all'impiego delle attrezzature che il progresso tecnologico pone a sua disposizione.

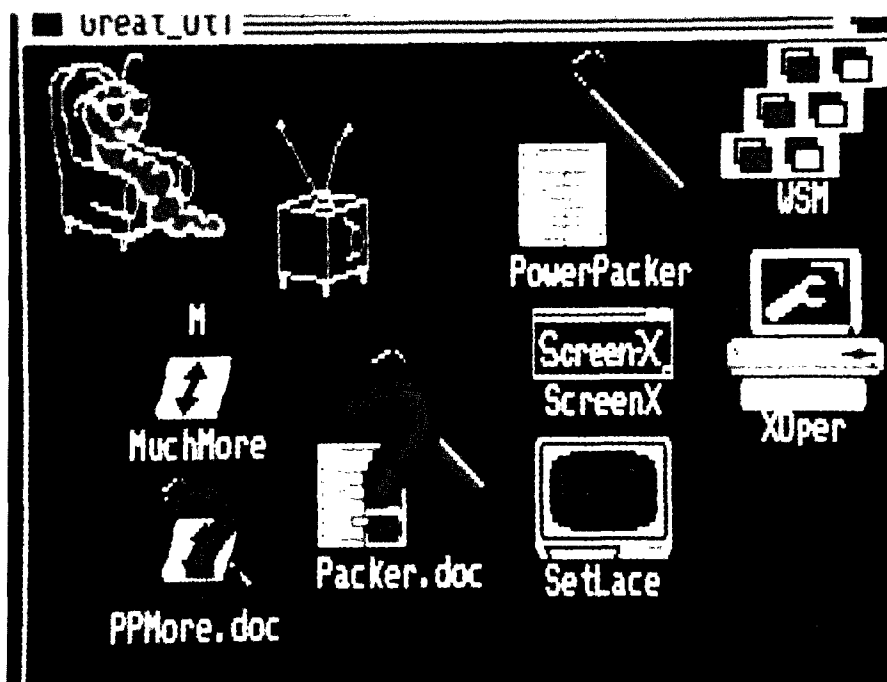
Per le ragioni sopraesposte, l'interazione tra strumenti elettronici e computer può rappresentare una risposta nell'automatizzazione di alcune procedure, specialmente nella registrazione di brani musicali, nell'editing di nuovi preset timbrici e nel trattamento di segnali campio-

nati. Il volume non è comunque dedicato ad uno specifico tipo di computer, ma vuole essere una sorta di introduzione alle potenzialità offerte da questa nuova disciplina. Amiga rappresenta senza dubbio uno dei computer emergenti in questo settore, specialmente per quanto riguarda l'ambito non professionale; molte software house stanno progettando per esso una grande quantità di software, quindi il futuro fa ben sperare in una sempre cresecente diffusione di questo computer.
Lo scopo del volume è quello di parlare di tecnica al musicista e non viceversa. Del MIDI vengono analizzate in dettaglio tutte le istruzioni e vengono forniti nel contempo esempi esplicativi ed applicazioni pratiche. Ampio spazio viene dedicato alla descrizione del funzionamento di apparecchiature MIDI quali: sintetizzatori, campionatori, sequencer. In particolar modo vengono trattate in dettaglio l'architettura e il funzionamento dei sintetizzatori digitali e dei campionatori, dei quali vengono fornite anche informazioni riguardanti le specifiche Hardware (struttura delle porte MIDI) e software utilizzate nella trasmissione di messaggi MIDI (Open Loop e Closed Loop ed in particolare i messaggi System Exclusive) oltre a trattarne in modo dettagliato le modalità di trasmissione e ricezione dei dati. Viene anche fornita una panoramica dei computer più diffusi in ambito musicale, unita ad una descrizione delle categorie software disponibili (sequencer, editor/librarian, notazione musicale, intelligenza artificiale ecc.) per il controllo e lo scambio di dati MIDI all'interno di un sistema composto da più apparecchiature. In particolare vengono esaminate le caratteristiche comuni alla maggior parte dei programmi appartenenti alla stessa categoria, indipendentemente dal tipo di computer utilizzato. Ad esempio, per quanto riguarda la descrizione dei sequencer MIDI, la traccia utilizzata prende come riferimento generale il programma Master Track Pro, di cui sono disponibili versioni per i computer: Apple Macintosh, Atari, Amiga, IBM PC e compatibili ed il recente Yamaha C1. Inoltre vengono trattati argomenti quali: Multitasking e Standard MIDI File dei quali vengono fornite struttura e specifiche. Infine vengono offerti suggerimenti sulle modalità di sincronizzazione (Sync to tape/sync from tape, SMPTE e MIDI Time Code) tra varie apparecchiature MIDI e sulla risoluzione dei problemi derivanti dal loro collegamento in rete (incompatibilità hardware e software, MIDI merge). Ampio spazio è dedicato alla rete MediaLink Lan, una delle future possibili soluzioni per potere espandere e migliorare le prestazioni del MIDI, senza ricorrere ad una modifica del progetto iniziale.
In ultima analisi il libro è orientato a fornire un valido supporto al musicista moderno in fase di composizione, di elaborazione del suono e di organizzazione di un efficiente sistema MIDI. ▲

Flicker Fixer

Contro lo sfarfallio dello schermo è arrivato Flicker Fixer !

Sebastiano Vigna

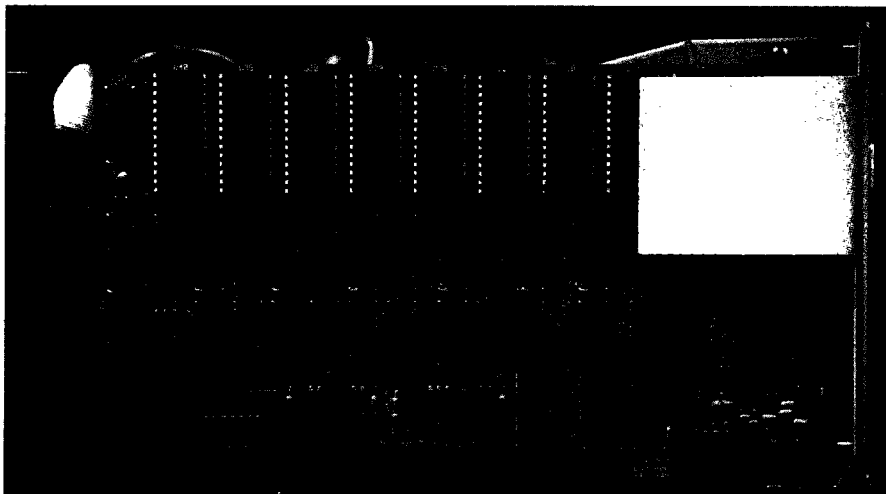
Chiunque possieda un Amiga, avrà certo inveito più volte contro il maledetto sfarfallio (più noto con il termine di flicker o flickering) che, in pratica, rende quasi inutilizzabili le risoluzioni interlacciate. Le specifiche tecniche di progettazione dei chip erano, ai tempi, di una grande semplicità nelle soluzioni per l'interfaccia con i sistemi video tradizionali, questi accettano senza problemi un segnale interlacciato a frequenza normale, ma richiedono speciali interfacce, se il segnale ha una frequenza superiore necessaria a "spedire" 512 linee per video frame. Purtroppo, l'handicap della mancanza di un modo a pixel più o meno quadrato che fosse stabile si è fatta sentire, in seguito, in modo particolarmente duro, anche come elemento di debolezza nei confronti dell'Atari ST (che però non ha avuto alcuna affermazione in campo video). Tuttora, il problema rimane aperto: ad esempio, per sfruttare un Word Processor grafico efficacemente è necessario poter visualizzare in modo corretto il Times, l'Helvetica e il Courier, ma questi font sono disegnati per un pixel quadrato, e si è quindi obbligati ad usare uno schermo interlacciato, con gli evidenti svantaggi visuali che ne conseguono.
Diverso tempo fa, una ditta americana, la MicroWay, ha prodotto una scheda per Amiga 2000 dal nome estremamente accattivante: Flicker Fixer, ovvero "tappa-sfarfallio". La scheda preleva direttamente dalla motherboard il segnale video e lo deinterlaccia, ovvero memorizza due videate e le trasmette in con-

Anche in bassa risoluzione, Flicker Fixer "riempie" i gap tra le scan line, ottenendo così una visualizzazione molto più nitida.

Le già notevoli immagini in Dynamic HiRes vengono esaltate da Flicker Fixer.

temporanea a frequenza maggiore. La prima condizione per l'uso del Flicker Fixer è dunque il possesso di un monitor MultiSync o perlomeno VGA, dato che i monitor Commodore attualmente in commercio possono ricevere solo frequenze "normali". Il risultato è veramente difficile da descrivere a parole. Se avete visto una scheda VGA delle più recenti su un buon monitor, potete farvene un'idea: stabilità assoluta, nitidezza perfetta, sparizione dei "gap" tra le linee in modo non interlacciato. Flicker Fixer, infatti, "deinterlaccia" il segnale anche quando questo non è interlacciato, eliminando così il fastidiosissimo effetto di "rigatura" che appare quando si usa Amiga con un monitor di qualità. Alla prima uscita di questa scheda, ci fu chi ne parlò affermando che la differenza tra la visualizzazione standard e quella di Flicker Fixer è paragonabile alla differenza tra notte e giorno, e il giudizio mi trova perfettamente d'accordo. La scheda vera e propria si inserisce con la massima facilità nello slot video di Amiga, e non richiede alcuna abilità per essere installata. Il segnale video "normale" è sempre disponibile, e quindi potete collegare due monitor contemporaneamente. L'ingenierizzazione è estremamente pulita, come del resto ci si può aspettare da una grossa casa come la MicroWay. Il manuale riporta, e qui bisogna veramente applaudire, tutti gli accoppiamenti che è necessario fare nella costruzione del cavo di collegamento tra Flicker Fixer e un qualunque monitor MultiSync (tutti o quasi tutti i modelli in commercio sono presi in considerazione). La disposizione dei pin del connettore sulla scheda è quella utilizzata dalla NEC, scelta giustificata dal fatto che proprio questa ditta ha introdotto per prima la tecnologia MultiSync.

La funzionalità della scheda è testimoniata dal suo enorme successo in America, ma solo in tempi relativamente recenti è stato realizzato un adattamento del Flicker Fixer ai canoni europei (come sapete, la versione PAL di Amiga ha 56 pixel in più). La nota dolente (almeno a parer mio) è però data dal fatto che, in apparenza allo scopo di non investire troppi soldi in un impresa di re-ingenierizzazione, la scheda americana è stata, per quanto possibile, "adattata" allo standard europeo: come conseguenza, il manuale (un fascio di fogli pinzato) afferma che non è possibile visualizzare più di 702x469 pixel, e che le dimensioni del Workbench vanno di conseguenza ridotte con il programma MoreRows (fornito con Flicker Fixer). Dopo un notevole numero di esperimenti ho potuto appurare che questo dato è falso, e per di più in senso negativo! In realtà, la scheda riesce a visualizzare 704x504 pixel senza alcuna difficoltà, perdendo quindi solo 8 pixel ad alta risoluzione rispetto alle dimensioni standard del Workbench. Per chi, come me, il Workbench è ormai fissato a 544 pixel verticali, la perdita è evidentemente più secca.

Un'altra piccola idiosincrasia che

Flicker Fixer montato nello slot video di Amiga.

può dare sorprese è la sincronizzazione che molti programmi dotati di smooth scrolling fanno con il segnale video, ad esempio MuchMore o CED. In genere, questi programmi spostano il video di un pixel per frame, in modo da ottenere un movimento perfettamente liscio. Il problema è che Flicker Fixer sovrappone due schermate consecutive, rendendo quasi illeggibile il testo durante lo spostamento. In pratica, tutte le situazioni di animazione corrono il rischio di non essere rese correttamente, ma dato che il segnale video standard è, come abbiamo detto, sempre disponibile, questa non mi sembra una difficoltà insormontabile.
Non ci si può rendere conto dell'incredibile utilità di Flicker Fixer (e di quanto sia una palla al piede l'interlacciamento) se non nell'uso quotidiano di Amiga. Non credo di esagerare affermando che si "vedono" le cose in maniera veramente diversa. Ad esempio, il Siesta 11, utilizzato da molti programmi in alta risoluzione interlacciata, sembra quasi un altro font. La retina ne ha una percezione molto più precisa, e in particolare non si avvertono più alcune spigolosità che, invece, si notano quando i caratteri si usano in modo interlacciato. Il previewer di AmigaTeX diventa veramente stupefacente, una volta che lo si può usare in un nero su bianco ben contrastato. E, come ho già accennato, anche i programmi che non girano in grafica interlacciata assumono un aria più nitida e, perché no, "professionale". E' un vero peccato che così pochi negozi abbiano in esposizione un Amiga dotato di Flicker Fixer: sarebbe un punto a nostro favore nella spietata lotta per il possesso del mercato.

Conclusioni

Non è facile tirare le fila di un discorso così complesso. Va preso in considerazione, inoltre, un altro fattore, ovvero la (sperata) imminente introduzione sul mercato dei nuovi chip grafici da parte della Commodore. Uno dei nuovi modi di visualizzazione sarà proprio un'alta risoluzione non interlacciata, e sebbene saranno disponibili solo quattro colori su 64, a chi fa del word processing o del desktop publishing non occorrerà certo di più. Per queste categorie di utenti potrebbe risultare più utile, a meno di urgenze immediate, attendere l'ECS (Enhanced Chip Set). Se, invece, (come ad esempio nelle applicazioni CAD) il colore è una componente essenziale, Flicker Fixer rimane una carta vincente, posto che si possa passare sopra alla mancanza di otto pixel. Prima di averlo si giura di non averne alcun bisogno, dopo l'acquisto si giura di non poterne fare a meno. ▲

Flicker Fixer è in vendita presso: Digimail s.r.l. - Via Coronelli, 10 Milano - Tel.02/426559

Un'immagine in SHAM in bassa risoluzione. Si noti anche in questo caso l'assenza di gap tra le linee.

European Computer Trade Show

Filippo Canavese

Dal 1 al 3 Aprile 1990 si è svolta a Londra la seconda edizione dell'European Computer Trade Show, una manifestazione che, nonostante sia così giovane, si è imposta come il principale avvenimento del mercato europeo nel settore consumer dei videogiochi e dell'Home productivity.
L'European Computer Trade Show, organizzato da Blenheim Database Exhibitions con la sponsorizzazione di CTW (Computer Trade Weekly) e ELSPA (European Leisure Software Publisher Association), ha visto la partecipazione di oltre 100 espositori che, nella cornice del caratteristico "Bussiness Design Centre", hanno potuto incontrare operatori del settore provenienti dall'Europa; dagli USA e dal Far East.
All'E.C.T.S. erano presenti tutti i settori del mercato consumer: dai produttori di hardware ai distributori, dalle software house ai produttori di periferiche, dalle riviste specializzate agli sviluppatori di software con la presenza dei più illustri nomi del mercato.
L'European Computer Trade Show 1990 è stato considerato, inoltre, come la principale rampa di lancio per le "collezioni" primaverili di offerte per il tempo libero e piccoli settori d'affari delle industrie produttrici di computer internazionali.
Sono stati presentati circa 200 nuovi giochi, il doppio del numero annunciato allo show inaugurale dell'anno scorso. Ancora una volta i videogiochi sono stati i protagonisti assoluti. La macchina leader è Amiga cui sono stati adattati 96 nuovi titoli, seguita da vicino da Atari ST con 87 e dal PC con 73.
Prima nella lista per i prodotti per Amiga è la software house francese Loriciel, che ha proposto 11 nuovi giochi: West Phaser completo della light gun, Crazy Shot, Moonblaster, Tennis Cup, Panza Kick Boxing, Superskweek, The Magician, Outboard, Sherman M4, Harricana e Pinball Magic.
La Microprose non ha voluto essere da meno e ci ha rivelato nove nuovi titoli: Mid Winter, Red Storm Rising, Pirates, First Contact, F-19, Tower of Babel, Fire and Brimstone, World Soccer '90 e RVF.
Nel frattempo, la Mirrorsoft ha proposto sette nuovi giochi: Gravity, Cadaver, The Final Battle, Theme Park Mystery, Back to The Future II: Paradox, TV Sports Basketball e Flight of The Intruder.
Per non rimanere fuori dal gioco, la Virgin Mastertronic, ci ha presentato cinque nuovi game: Dan Dare III: The Return, World Cup Soccer '90 e su etichetta di una sua affiliata, la 16 Blitz, Grimblood, Conflict e Hunter Killer.
Una delle game house leader del mercato mondiale, la Psygnosis, ha annunciato allo show i preview che verranno realizzati durante tutto il 1990. Tra gli altri: Anarchy, Firestone, The Keep, Barbarian II, Beast II, Gore, Tempus e Turbo Buggies.
Ecco un sostanzioso elenco di alcuni dei giochi previsti per Amiga: Treasure Trap, Xiphos (Electronic Zoo), Orcus, Prophecy I: The Vicking Child (Wired), Powerboat USA (Accolade), No Exit (Coktel Vision), European Superleague (CDS), Laurel & Hardy, The Last Stuntman, International Athletics, Sorceror's Apprentice, Search For The Titanic (CRL), Colditz, Rotate (Digital Magic), Manchester United, Rogue Trooper (Krisalis), Imperium (Electronic Arts), Pipemania, Space Ace, Dragon's Lair: Escape from Singe's Castle, Paul Gascoigne Soccer,

Mark Lewis, Managing Director dell'Electronic Arts, e Piero Dell'Orco, Direttore responsabile di BIT

Lo stand Genias ospitata dalla Linel

Team Yankee (Empire), Wild Street, Dark Sentry, Crazy Bikes, Fire & Forget II (Titus), Venus, BSS Jane Seymour, Combo Racer (Gremlin Graphics), Paradroid '90, Scavenger, Nebulus II (Hewson), Xenomorph, Debut (Interceptor), Damocles (Novagen), Circle of Blood (The Edge), The Last Starship, Rourke's Drift, Web of Terror (Impressions), Voodoo Nightmare, Operation Stealth (Palace), Colorado, Strablade, Maya (Silmarils), Last Dutchman's Mine, Persian Gulf Inferno, USS John Young (Interprize), Elvira: Mistress of The Dark, H20, I Was Kidnapped By a Flying Saucer (Tynesoft), Jumping Jackson, Fatal Gravity, The Toyottes (Infogrames), Thomas The Tank Engine (Alternative), Castle Master (Incentive), Klax, Escape From The Planet of The Robot Monsters, Cyberball (Tengen), The Spy Who Loved Me (Domark), Yogi's Great Escape, Hong Kong Phooey No. 1 Super Guy (Hi-Tec).
Sul fronte dei "non-giochi", nuove utilitiy previste per Amiga sono: Zooram che si completa con cinque giochi della Electronic Zoo; Prodata, un nuovo database della Arnor; Music For Junior, della MicroIllusions; Deluxe Paint Tutorial, un set di tutoriali per disegnare e animare, edito dalla US Action.
"Ancora una volta l'European Computer Trade Show, ha dato prova di essere una rampa di lancio per i nuovi prodotti", afferma Michael Meakin, direttore della Blenheim Database Exhibitions, organizzatrice dello Show.
"Il fatto che il numero dei lanci sia raddoppiato quest'anno, rispecchia la crescita di questo mercato". Tra gli espositori americani, le migliori Case come la Lucasfilm di San Rafael, California o gli specialisti in periferiche, la Suncom di Niles, Illinois, che hanno scelto lo Show per la prima esibizione dei loro sintetizzatori musicali fuori dagli USA.
La Innerprize di Chicago è stata rappresentata dalla Casa The Software Business così come la MicroIllusions. Altre compagnie americane si sono presentate direttamente o tramite le filiali europee, tra le quali: Sub Logic, Accolade, Sierra on Line, Bethesda, Broderbund, Electronic Arts, Micropose, Mindscape ed Electronic Zoo.
Rivista specialistica degli States presente allo Show è la Software Publisher's Association.
"E' un evento molto importante per fare affari", ha affermato il rappresentante ufficiale Cheryl Davis. "Non rappresentiamo i colonizzatori americani che arrivano per conquistare l'Europa. Questo è un biglietto di andata e ritorno con il quale possiamo aiutare le compagnie europee ad arrivare bene negli States".
Il Giappone, gigante delle console, è stato rappresentato dalla Serif Games e a contenere il dominio della Gran Bretagna, come unica rappresentante dell'Europa, sono arrivate Germania, Francia, Spagna, Italia, Svizzera e Scandinavia. Le Case Ariolasoft, EAS, Profisoft, Rushware e Starbyte costituivano il contingente della Germania.
Nell'avanguardia della delegazione francese erano presenti la Coktel Vision, la Infrogames, la Jong Riche, la Loriciel, la Silmarils e la Titus. Per la Svizzera la Liner e per l'Italia, oltre ovviamente alla presenza del Gruppo Editoriale Jackson, la Leader Distribuzione e in abbinamento con la Linel la neonata software house italiana Genias.
Nel corso dei prossimi mesi avremo modo di recensire più dettagliatamente molti dei prodotti dell'ECTS '90. Diamo ora uno sguardo all'avvenimento clou della manifestazione e cioè alla "notte degli Oscar".

European Computer Leisure Awards '90

Commodore, Citizen ed Electronic Arts sono stati nominati i vincitori del primo "European Computer Leisure Awards '90" nel corso della premiazione tenutasi il 1 Aprile.
Amiga ha ricevuto il premio come computer dell'anno nella sezione Home Productivity, mentre la Citizen Swift 24 si è portata via il titolo di stampante dell'anno.
Electronic Arts ha ricevuto ben quattro premi: due nella sezione tempo libero, con Populous, gioco dell'anno, e come "software publisher" dell'anno; e due nella sezione "Home Productivity" con il programma Deluxe Paint III quale miglior programma per Amiga e miglior programma di computer grafica.
L'assegnazione dei premi, in perfetto stile serata da Oscar, si è tenuto nella splendida Empire Suite del Cafe Royal in Regent Street, uno dei pochi locali rimasti indenni dopo i violenti scontri avvenuti il giorno precedente in seguito alla manifestazione indetta contro la "Poll Tax".

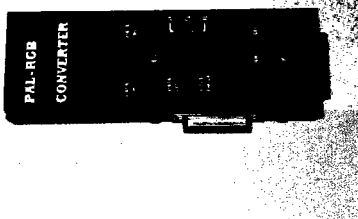
Circa 250 dirigenti delle società leader del settore hanno partecipato alla cena di premiazione.
La giuria era composta dai rappresentanti di 19 riviste europee, Statunitensi e giapponesi; e più specificamente: Amiga Format, ST Format, New Computer Express, Amstrad Action, Your Sinclair per la Gran Bretagna; Tilt, PCompatibles, Generation 4 per la Francia; Computer Live, Power Play per la Germania; Guida Video Giochi, PC Software, **Amiga Magazine** e Bit per l'Italia; Micro Mania, Amstrad Professional per la Spagna; Dator Magazine per la Svezia; Compute! per gli Stati Uniti e LogIn per il Giappone.
Il primo giudizio ha visto una serie di competizioni nazionali, con un team di giornalisti di ogni Paese che hanno selezionato chi avrebbe partecipato al gran finale.
Questo è stato fatto in totale segretezza il pomeriggio prima della cerimonia di premiazione.
Anfitrione della serata è stato Lance Percival (famoso personaggio televisivo inglese) che ha premiato i vincitori.
Ecco i premi: Space Ace, Entertainment International premio per la miglior animazione; Shadow of The Beast, Psygnosis premio per la miglior grafica; Future Wars, Delphine/Palace Software premio per i migliori effetti sonori; Xenon II, Imageworks/Mirrorsoft premio per il miglior Shoot'em-up; Batman, Ocean Software premio per il miglior gioco d'azione; Rainbow Islands, Ocean Software premio per la miglior conversione dall'arcade; Indiana Jones and the Last Crusade, US Gold premio per il miglior gioco d'avventura; Chaos Strikes Back, FTL/Mirrorsoft premio per il miglior gioco di ruolo; Populous, Electronic Arts premio per la miglior strategia, per il gioco più originale, per il gioco dell'anno; F-16 Combat Pilot, Digital Integration premio per il miglior gioco di simulazione; Kick Off, Anco premio per la miglior simulazione di sport; Sim City, Infogrames premio per il miglior soft educativo; Electronic Arts, Casa Editrice dell'Anno; Commodore Amiga, Computer dell'Anno; Citizen Swift 24, Stampante dell'Anno; Norton Utilities 4.5, Peter Norton Computing,Utility dell'Anno; Animator, Autodesk premio per i migliori prodotti per PC; Spectra GCR premio per il miglior prodotto per ST; Deluxe Paint III, Electronic Arts premio per il miglior prodotto per Amiga e per la miglior grafica; Wordperfect 4 premio per il miglior Word Processor; Dbase IV, Ashton Tate premio per il miglior Database e Pagemaker, Aldus premio per il miglior DTP package.
Le pubbliche relazioni relative sia alla mostra sia all'Awards sono state egregiamente curate dalla CCA (Cape Cowley Associates). ▲

PAL-RGB Converter

Stefano Paganini

Il prodotto in prova questo mese è abbastanza particolare e rivolto ad un determinato tipo di utenza. Tutti gli utenti di Amiga conoscono i digitalizzatori video tipo Digi-View, chi ne possiede uno sa quanto sia problematico, nella maggior parte dei casi, effettuare correttamente la digitalizzazione in modo HAM.
La diffusione dei digitizer di tipo statico ha portato ad alcune soluzioni talvolta ingegnose, talvolta macchinose: è il caso del Digi-Droid della stessa NewTek, che in unione al Digi-View permette di cambiare i filtri tramite il collegamento di un motorino fissato sull'obiettivo della telecamera controllato dalla porta Joystick #2.
Il problema dei tre filtri cromatici è quantomai insolvibile nel caso di digitalizzazioni da VCR in fermo immagine. La soluzione viene da questo PAL-RGB Converter della P.G. Elettronica di Torino.
Questo modulo può coesistere con un qualunque digitalizzatore video, ad esempio il Video Digitizer della stessa P.G.
Il Converter presenta un connettore per il collegamento con la porta video di Amiga, e il medesimo in uscita al monitor; sono presenti due jack RCA per l'ingresso e l'uscita in videocomposito, due interruttori per la selezione del modo di funzionamento e due trimmer per la regolazione di contrasto e saturazione. Nella confezione sono compresi il manuale delle istruzioni e un cavetto video di circa 15 cm.
Una volta completati i collegamenti, il selettore Video Ext./AMIGA permette di passare dallo schermo AMIGA all'immagine del VCR o della telecamera, senza dover intervenire sul monitor. Ciò permette di posizionare la telecamera o il nastro sul punto esatto per poi digitalizzare. Quest'ultima operazione si svolge nel modo seguente: al posto di cambiare i filtri è sufficiente selezionare R-G-B per ottenere la separazione cromatica del segnale corrispondente. Il digitizing procede dunque senza problemi sia che si stia operando con una telecamera che con il VCR in fermo immagine.
I trimmer di regolazione, come specificato nel manuale, non vanno modificati, a meno di particolari disturbi, in quanto tarati su valori medi per qualsiasi segnale Videocomposito; il segnale in uscita è, comunque, migliorato nella maggior parte dei casi.
Una sola nota riguarda la scelta operata nel porre il connettore di uscita per il monitor su un lato del modulo: ciò ne pregiudica il funzionamento con Amiga 2000 a meno di particolari acrobazie tra cavi e prolunghe del connettore della porta parallela adiacente e, impiegata dal digitalizzatore video.
Nessun problema, invece, per Amiga 500. Ricordando che tale unità, al pari di tutte le periferiche, va inserita/disinserita SOLO a computer

spento, concludiamo sottolineando che il PAL-RGB Converter è un'ottima soluzione al problema di cui sopra: immagini in still frame da VCR, senza filtri e utilizzo della telecamera senza filtri.
E' solo auspicabile una versione per 2000, magari con una disposizione più accessibile dei comandi di selezione per una maggior praticità. ▲

Pal-RGB Converter è prodotto e distribuito da: P.G. Elettronica s.n.c. Via Garessio, 48/1 (TO) Tel. 011- 632549

Come utilizzare il dischetto

Prima di utilizzare il dischetto allegato alla rivista assicuratevi di averlo protetto dalla scrittura aprendone la finestrella, per evitare accidentali cancellazioni o contaminazioni da virus (il disco allegato è al 100% privo di virus ed il Gruppo Editoriale Jackson non si assume nessuna responsabilità in caso di contagio). A questo punto accendete il computer. Dopo la presentazione, dalla quale è possibile uscire premendo il tasto sinistro del mouse, verrà caricato il Workbench. Aprendo l'icona del disco (cliccando due volte con il tasto sinistro su di esso) appaiono i "cassetti" relativi a giochi, utilities, ecc. Aprite allo stesso modo il "cassetto" al quale siete interessati e caricate il programma desiderato. Alcuni programmi possono essere anche caricato da CLI con altre opzioni (vedere a parte le spiegazioni dei programmi). Per chi possiede Amiga con "soli" 512 K di memoria RAM è consigliabile chiudere tutte le finestre aperte tranne quella relativa al programma da far girare (che può essere eventualmente chiusa con un po' di prontezza di riflessi durante il caricamento). In tal modo il programma avrà a disposizione più memoria. In caso contrario, potrebbero verificarsi dei malfunzionamenti.
ATTENZIONE: prima di far partire ogni programma vi raccomandiamo di leggere attentamente le istruzioni del programma. I programmi Maze, Turbo, StealMemBoot, PopInfo e idue programmi-scherzo sono di pubblico dominio, e possono essere liberamente distribuiti. Inoltre, ricordiamo che, i programmi sorgenti ed eseguibili delle pagine di TRANSACTOR per AMIGA, si trovano nel cassetto TRANSACTOR e devono essere caricati da CLI. Tutti gli altri programmi presenti sul dischetto sono, invece, coperti da Copyright 1990 Gruppo Editoriale Jackson-Compute! Publications Inc., e come tali non devono essere copiati e distribuiti senza la preventiva autorizzazione scritta dell'editore. Amiga Workbench 1.3 è copyright 1985, 1986,1987,1988 Commodore-Amiga Inc., tutti i diritti sono riservati.
Per ogni questione tecnica riguardante il funzionamento del disco telefonare il Martedì e il Giovedì dalle ore 14.30 alle ore 16.00 al 02-6948287.

Come entrare in CLI

- Accendere il computer
- Inserire il disco con il Workbench
- Attendere il caricamento
- Cliccare due volte sul disco del Woorkbench
- Cliccare due volte sull'icona SHELL (per il nuovo CLI), oppure aprire il il cassetto System
- Selezionare l'icona CLI

Esempi di caricamento da CLI
- Entrare in CLI
- Digitare: COPY FROM C: TO RAM:
- Inserire il disco di Amiga Magazine e digitare CD DF0:
- Digitare CD UTILITIES

A questo punto, per esempio, si può digitare:

- PopInfo
- Turbo

Amiga Magazine n.13

SERVIZIO LETTORI **Compilare e spedire in busta chiusa a: GRUPPO EDITORIALE JACKSON Area Consumer - Via Pola, 9 - 20124 Milano**

A) Come giudichi questo numero di Amiga Magazine ?

- ❑ Ottimo
- ❑ Molto Buono
- ❑ Buono
- ❑ Discreto
- ❑ Sufficiente
- ❑ Insufficiente

B) Quale (i) articolo (i) o rubrica hai apprezzato di più?

Quale meno?

C) Cosa ti piacerebbe leggere nei prossimi numeri di Amiga Magazine ?

D) Ti è piaciuto il Software ?

E) Quante persone leggono la tua copia di Amiga Magazine ?

F) Possiedi un computer? Quale?

Quale (i) computer intendi acquistare in futuro?

G) Leggi altre riviste Jackson? ❑ SI ❑ NO

Quali?

H) Leggi altre riviste del settore? ❑ SI ❑ NO

Quali?

I) Oltre alle riviste dedicate al computer quali sono le tue letture preferite?

L) Quali sono i tuoi hobbies e maggiori interessi?

- ❑ Sport
- ❑ Musica
- ❑ Videoregistrazione
- ❑ Hi - Fi
- ❑ Fotografia
- ❑ Automobile
- ❑ Moto
- ❑ Viaggi

Nome
Cognome
Indirizzo
Età **Professione**
Città
Prov. **C.a.p.** **Tel.**

a cura di G. Biagini e A. Laus

Nota dei voti

Grafica
Tiene conto della cura nel disegno dei fondali e degli sprite, dell'animazione, dello screen dei titoli.

Sonoro
Valuta la colonna sonora e gli effetti sonori.

Giocabilità
Tiene conto della varietà del gioco, delle opzioni e la possibilità di giocare in due e di eventuali scelte.

Durata
Indica per quanto tempo ci giocherete.

Generale
E' il voto generale dei valori precedenti e di tutto ciò che accompagna il gioco: confezione, manuale, gadget, ecc.

I voti vengono espressi in percentuale da 1 a 100.

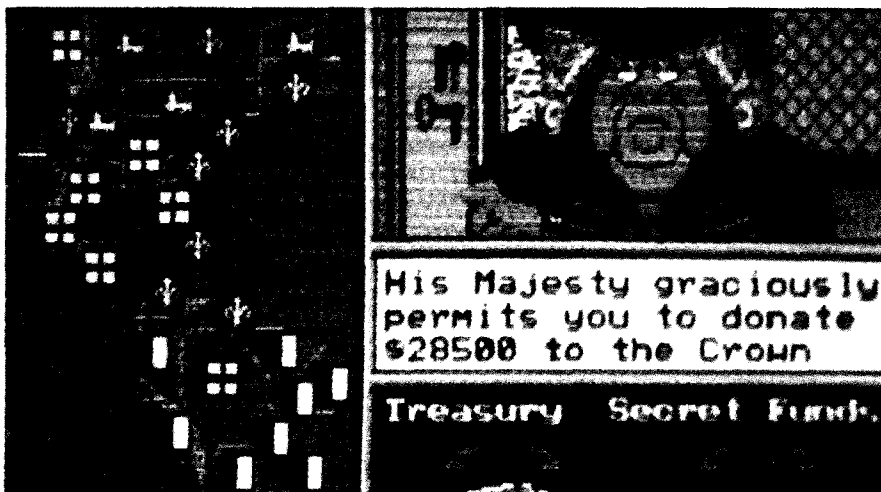
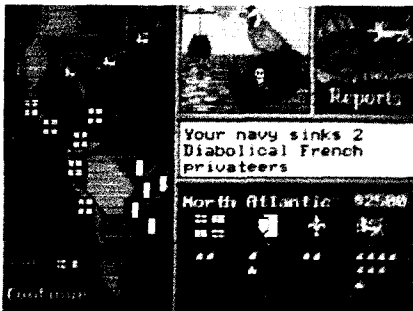
Golds of the Americas

Grafica	87
Sonoro	70
Giocabilità	90
Durata	89
Generale	89

Software house: S.S.G.
Prezzo: 79.000 lire

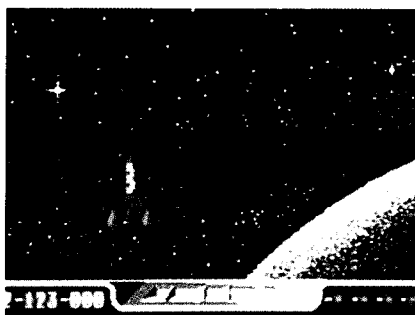
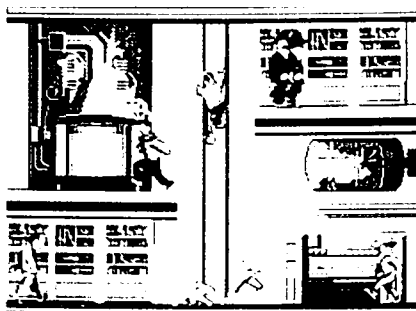
Appassionati di giochi Roll Bar o appassionati di storia, ecco un gioco che fa per voi: Golds of America è una simulazione di colonizzazione fatta molto bene dove si può giocare da uno a quattro giocatori contemporaneamente (se si gioca da soli gli altri tre partecipanti sono simulati dal computer). L'inizio del gioco è ambientato nel 1510: quattro nazioni sono in lizza per il predominio del territorio americano, le più grandi potenze europee del momento: Francia, Spagna, Portogallo e Inghilterra si fronteggiano in battaglie navali e in conquiste territoriali spesso guidate da esploratori passati alla storia come Pizarro, Vespucci e così via. E' possibile scegliere il territorio americano seguendo le spe-

cifiche indelebilmente dettate dalla storia le quali stabiliscono che i Portoghesi si spingano in Sud America e i Francesi al nord o addirittura creare un nuovo territorio. Il gioco, soprattutto a quattro, si fa subito interessante: ogni giocatore può disporre di una serie di risorse da gestire al meglio, si possono creare delle colonie attraverso le esplorazioni sempre più interne del paese come le si possono conquistare ad altri paesi con i quali si è in lotta, si possono stringere alleanze, dichiarare guerre (da notare che queste sono opzioni che vengono automaticamente scelte dal computer tenendo conto delle date storiche che si stanno vivendo).
Ottimo prodotto dunque questo Golds of the Americas: nello svolgimento del gioco si può gestire il proprio predominio e la propria sete di potere con l'aiuto di pirati che vanno a distruggere i vascelli mercantili, potenziare le riserve economiche del paese attraverso l'importazione di mano d'opera di colore, riempire gli oceani di navi da combattimento fedeli alla propria bandiera e, infine, incrementare lo sviluppo demografico attraverso l'insediamento di coloni i quali cercheranno di tenere stretta la terra a loro concessa e si spingeranno, aiutati dagli esploratori, alla conquista di nuove terre. Il gioco è effettivamente complicato, ma la posta in gioco è alta: si parla di essere padroni del nuovo continente e scusate se è poco!

Golds of the Americas è distribuito da: C.T.O. s.r.l. - Via Piemonte, 7/F 40069 Zola Predosa (Bo) tel.051/753133

Tin Tin on the Moon

Grafica	86
Sonoro	70
Giocabilità	65
Durata	75
Generale	78

Software house: Infogrames
Prezzo: 39.000 lire

Quando ero piccolo c'era in televisione Supergulp, una trasmissione che proponeva i cartoni animati di TIN TIN, il famoso eroe dei fumetti francesi. Sicuramente la Infogrames, forse la più nota software francese in campo Amiga, ha avuto un'ottima idea a presentare il videogioco del cartone di cui sopra.
I personaggi ci sono proprio tutti, Tin Tin, il cagnetto Milou, gli investigatori gemelli Dupondt e il comandante Haddock. Le loro funzioni sono ben definite nello svolgersi dell'azione: il capitano, per esempio, riesce a frenare il cattivo colonnello Jorgen, Milou sveglia Tin Tin quando è svenuto, i gemelli Dupondt tengono sulla retta via l'ingegner Wolf che a sua volta guida Tin Tin verso le bombe da disinnescare. L'azione è quindi molto complessa: all'inizio si parte con il razzo alla volta della Luna (qui vedrete una presentazione splendida soprattutto per il cambio di inquadratura di notevolissimo effetto); dopo essere atterrati si devono disinnescare una serie di bombe e, infine, si deve catturare il colonnello Jorgen saltandogli sulla schiena e immobilizzandolo. Il gioco è interessante e molto ben curato, ma il problema è il manuale, tanto esiguo da rendere il gioco un coacervo di byte inutilizzabili.
La C.T.O. fornisce un manualetto composto da una pagina sulla quale sono riportati comandi che non esistono (per esempio viaggiando nello spazio non si può sparare quantunque il "manuale" dica di si), non compaiono le istruzioni relative al disinnesco delle bombe né nulla che possa aiutare minimamente a capire il gioco. C'è, è vero, anche un foglio aggiuntivo, che è comunque troppo esiguo per spiegare come utilizzare il programma (peraltro interessante). A voi scoprire le istruzioni, allora, perché il gioco è effettivamente piacevole.

Tin Tin on the Moon è distribuito da: C.T.O. s.r.l. - Tel. 051/753133

Galdregons Domain

Grafica	80
Sonoro	75
Giocabilità	78
Durata	70
Generale	75

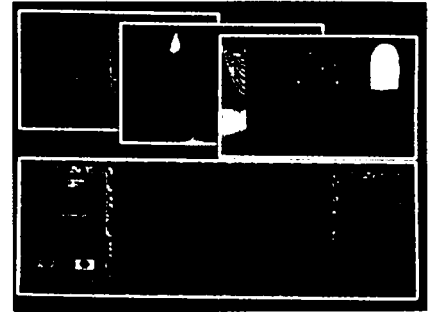
Software house: Pandora
Prezzo: 39.000 lire

Un bel gioco, ambientato nel mondo della fantasia, in cui voi sarete nientemeno che un fortissimo barbaro che, alla ricerca di fama e fortuna arriva alla corte del re Rohan.
Il re gli ordina di cercare le leggendarie cinque gemme dai poteri magici, sparse per tutto il regno e in mano a creature fortissime e cattivissime.
Qui comincia la vostra avventura che, lo dico subito, non è per niente facile.
Conoscete l'ubicazione di una sola delle gemme che, naturalmente, deve essere presa per ultima (che gusto ci sarebbe altrimenti?)
Il vostro arduo compito è appesantito dalla responsabilità che vi siete accollati: se fallirete, il mondo sarà

ridotto in schiavitù.
Per trovare le gemme dovrete fare molta strada e raccogliere oggetti che potrebbero darvi informazioni utili al ritrovamento; le varie locande, sparse per tutto il territorio sono un'ottima fonte di informazioni.
Strada facendo potrete imbattervi in armi che, una volta raccolte, si riveleranno molto utili nel combattimento.
Voi siete molto più forti di tutte le creature che incontrerete e se saprete combattere bene, non avrete problemi a sconfiggerle; in ogni caso non tentate di uccidere tutti quelli che incontrate: il vostro scopo è di trovare le gemme, non di fare il killer.
Per i movimenti utilizzate il mouse; l'azione del gioco è controllata dal posizionamento del pointer sulle varie icone presenti sullo schermo.
La schermata principale è divisa in due parti: in alto c'è l'azione vera e propria, mentre in basso ci sono le varie icone: quelle per il movimento, quella del combattimento e quelle relative agli oggetti che prendete per strada.
Nonostante l'apparente macchinosità dell'uso dei comandi per i movimenti, il gioco è fluido e scorrevole; il difficile è trovare le gemme che sono davvero molto ben nascoste e protette.
Nella confezione, oltre al manuale, in italiano e in inglese e ai due dischetti del programma, c'è anche un poster del gioco.

Galdregons Domain è distribuito da: Softel
Via E. Dal Pozzo, 7
00146 - Roma
Tel. 06/5584334

Mystere

Grafica	89
Sonoro	75
Giocabilità	90
Durata	88
Generale	88

Software house: Genias
Prezzo: 39.000 lire

Mystere la fuga dal castello di Lockness, della nuova software house italiana, è un adventure grafico sullo stile di Lancelot e il risultato non è niente male. Il tutto comincia con lo sfoglio di una pergamena sulla quale un vostro nonno ha lasciato il suo testamento: tutti i suoi averi saranno vostri a condizione che riusciate a trovare il quadro del mostro del lago di Lockness che la leggenda vuole nascosto nei meandri del castello del lago. Il vostro temperamento vi spinge ad andare alla ricerca del dipinto al quale sono vincolate le sostanze del vecchio, ma subito si percepiscono i primi segni dell'alone di mistero che circonda il maniero: il barcaiolo che vi ha condotto sul luogo vi abbandona pur di non passare la notte nella magione, e voi vi trovate solo contro tutti. Tralasciando la trama vediamo

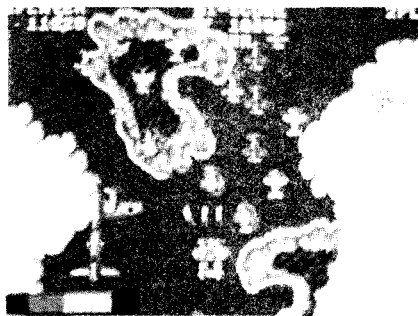
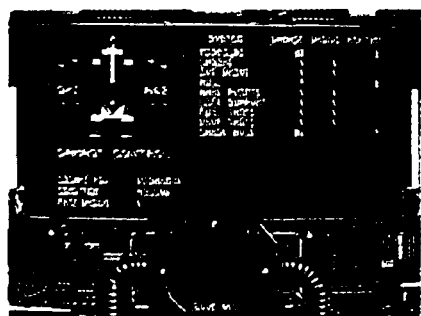
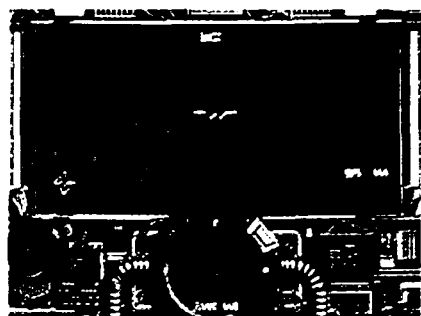
subito quali sono i punti caratteristici dell'adventure: innanzitutto vi è un interprete delle frasi inserite dall'utente che ha delle ottime potenzialità; attua un controllo semantico del costrutto da voi inserito (che può raggiungere gradi di complessità abbastanza elevati) e lo interpreta in maniera molto lineare, quindi è possibile scrivere frasi più descrittive che nelle altre adventure. La grafica è ben curata: è possibile scegliere i comandi rappresentati da icone o addirittura quelli in italiano (da notare che tutta l'adventure è fatta nella nostra lingua madre). In conclusione un prodotto che come dice il manuale stesso non ha niente da invidiare a quelli esteri, un buon passo avanti per le software house italiane che stanno iniziando ad affacciarsi timidamente sul mercato Amiga.

Mystere è prodotto da: Genias
Via A.Volta, 2/A
40055 Castenaso (Bo)
Tel.051/784305

Federation of Free Traiders

Grafica	88
Sonoro	85
Giocabilità	82
Durata	81
Generale	84

Software house: Gremlins
Prezzo: 59.000 lire

Ecco quel che si dice un gioco interessante che riesce a conglobare le caratteristiche di un arcade con la classe e l'interesse di un gioco di strategia. Un gioco che tiene conto dell'intelligenza del giocatore e che, malgrado la sua complessità, riesce ad appassionare e ad essere apprezzato. Iniziamo a parlare del manuale che è composto di più parti, uno dei libri contiene la storia

del gioco che merita una nota di attenzione: il mondo in cui vi trovate è animato da floridissimi scambi commerciali tra pianeti, voi siete solo un novellino, ma avete un'astronave e decidete di iniziare ad entrare in questo giro e, forte di una sovvenzione economica regalatavi da uno sconosciuto sostenitore, iniziate ad entrare nel favoloso mondo della "Federazione dei liberi mercanti". Tralasciando la trama (stupenda peraltro) vediamo le particolarità del gioco: un universo nel quale spaziare, di dimensioni estesissime, si possono visitare oltre otto milioni di pianeti distribuiti su uno spazio di circa 18 trilioni di anni luce; un linguaggio di programmazione del computer di bordo simil Basic, un linguaggio di comunicazione tra astronavi di tipo evoluto e, infine, una grandiosa grafica vettoriale.
Come vedete i buoni presupposti ci sono tutti e vi posso assicurare che FOFT è un programma veramente appassionante; unico piccolo neo è la poca interattività con gli scambi di borsa, cosicché è difficile acquistare e vendere ricavando denaro dal plusvalore delle merci acquisite in precedenza.
Una ottima colonna sonora (che la Gremlins, casa produttrice, propone anche registrata su una musicassetta inserita gratuitamente nella confezione) accompagna le stupende scene in grafica vettoriale. Da notare anche lo stupendo demo all'inizio del gioco.

F.O.F.T. è distribuito da:
Leader Distribuzione
Via Mazzini, 15
Casciago (Va)
Tel. 0332/212255

1943

Grafica	75
Sonoro	67
Giocabilità	70
Durata	69
Generale	77

Software house: Capcom
Prezzo: 18.000 lire

Ricordate il bombardamento di Pearl Harbour? Fu quello che spinse gli Stati Uniti d'America ad entrare in guerra contro il Giappone, impero del Sol Levante. La guerra iniziò e ci fu una battaglia che decise le sorti dello scontro, questa fu la battaglia delle Midway; sta a voi far rivivere al vostro Amiga le scene di quell'episodio.
Scopo del gioco (ed anche della battaglia originale) è quello di distruggere l'incrociatore giapponese Yamato. Si ha a disposizione un aereo e una serie di armi bonus che si guadagnano raccogliendo i simboli POW che compaiono sullo schermo. State pensando di aver già visto un gioco del genere al bar? Ebbene sì, questo della Capcom è il coin up della versione da bar.
Graficamente non convince molto, i colori sono troppo eterogenei, gli scenari sono poco definiti, si direbbe che il gioco sia stato fatto con un Commodore 64, può piacere o non piacere, comunque non è, a mio parere, un gioco da Amiga.
Certo se amate l'azione frenetica vi potete divertire; sono tra l'altro disponibili diversi tipi di sparo e, a seconda del grado di bravura, ci si può spingere molto oltre, ma manca quel qualcosa che lo renda affascinante. Rimane nella categoria degli shoot'em-up ma è frammentario, insomma, era molto meglio quello del bar.

1943 è distribuito da: Leader Distribuzione - Tel. 0332/212255

COMPILARE E SPEDIRE IN BUSTA CHIUSA A: GRUPPO EDITORIALE JACKSON - VIA ROSELLINI 12 - 20124 MILANO

ABBONAMENTO RIVISTE JACKSON

ATTENZIONE La cedola di richiesta abbonamenti presenta un modulo speciale con una serie di domande a cui preghiamo vivamente di rispondere con precisione.
AVVERTENZA I rinnovi entreranno in vigore automaticamente a partire dal numero successivo alla data di scadenza dell'abbonamento precedente. Per motivi tecnici, i nuovi abbonamenti saranno attivati dopo 6 settimane dalla data di ricevimento della cedola di richiesta, o in data successiva nel caso di esplicita comunicazione da parte dell'abbonato.

N.B. Per abbonamenti all'estero le tariffe dovranno essere raddoppiate. Non è prevista la spedizione via aerea

❑ NUOVO ABBONAMENTO ❑ RINNOVO

Rivista	Numeri	Prezzo
❑ BIT	numeri 11	L. 53.000
❑ PC MAGAZINE	numeri 11	L. 52.000
❑ PC FLOPPY	numeri 11	L. 105.500
❑ INFORMATICA OGGI	numeri 11	L. 52.500
❑ INFORMATICA OGGI SETTIMANALE	numeri 40	L. 32.000
❑ TRASMISSIONE DATI E TELECOM.	numeri 11	L. 51.500
❑ COMPUTERGRAFICA & DTP	numeri 11	L. 51.000
❑ NTE COMPUSCUOLA	numeri 10	L. 33.000
❑ ELETTRONICA OGGI	numeri 20	L. 112 000
❑ AUTOMAZIONE OGGI	numeri 20	L. 80.000
❑ EO NEWS SETTIMANALE	numeri 40	L. 32.500
❑ STRUMENTAZIONE E MISURE OGGI	numeri 11	L. 53.500
❑ MECCANICA OGGI	numeri 11	L. 61.500
❑ STRUMENTI MUSICALI	numeri 11	L. 50.500
❑ WATT	numeri 20	L. 31.500
❑ FARE ELETTRONICA	numeri 12	L. 58.000
❑ AMIGA MAGAZINE (disk)	numeri 11	L. 123.500
❑ SUPERCOMMODORE (disk)	numeri 11	L. 110.000
❑ SUPERCOMMODORE (tape)	numeri 11	L. 75.000
❑ PC SOFTWARE 5 1/4	numeri 11	L. 106.000
❑ PC SOFTWARE 3 1/2	numeri 11	L. 132 000
❑ PC GAMES 5 1/4	numeri 11	L. 124.000
❑ PC GAMES 3 1/2	numeri 11	L. 132.500
❑ GUIDA VIDEOGIOCHI	numeri 11	L. 31.000

MODALITA' DI PAGAMENTO

Allego assegno n. ______ di L ______
Banca ______
Ho effettuato versamento di L. ______ sul c/c postale n.11666203 intestato a Gruppo Editoriale Jackson-Milano e allego fotocopia della ricevuta
Addebitate l'importo di L. ______ sulla carta di credito
AMERICAN EXPRESS
BANKAMERICARD
CARTA SI'
DINERS CLUB
Data di scadenza della carta di credito ______
Data ______ Firma ______

JACKSON CARD 1990

GRUPPO EDITORIALE JACKSON

Nuova, più ricca e sempre più preziosa

Ecco tutti i vantaggi che ti offre la Jackson Card 90: ● sconti esclusivi e facilitazioni negli oltre 3000 punti di vendita convenzionati ● sconto speciale del 10% sull'acquisto di libri Jackson ● invio gratuito del catalogo libri 1990 e aggiornamento sulle ultime novità editoriali ● invio gratuito della rivista Jackson Preview Magazine ● buono acquisto speciale di L. 15.000 su ordini di libri effettuati per corrispondenza.

Ottenerla è facile.........

CEDOLA DI COMMISSIONE LIBRARIA

Si inviatemi i volumi sotto elencati:
❑ All'indirizzo privato ❑ All'indirizzo di lavoro

INDIRIZZO PRIVATO

COGNOME E NOME ______
VIA E NUMERO ______
CAP ______ CITTA' ______ PROV. ______
TEL. (______) ______ ANNO DI NASCITA 19 ______
TITOLO DI STUDIO. ❑ LAUREA ❑ MEDIA SUPERIORE ❑ MEDIA INFERIORE

INDIRIZZO LAVORO

DITTA O ENTE ______
VIA E NUMERO ______
CAP ______ CITTA' ______ PROV. ______
TEL. (______) ______ TELEX/FAX ______

INDICARE CHIARAMENTE CODICI E QUANTITA' DEI VOLUMI RICHIESTI

CODICE	Q.TA	CODICE	Q.TA	CODICE	Q.TA	CODICE	Q.TA

❑ Non sono titolare della Jackson Card
❑ Sono titolare della Jackson Card '90 n° ______
e ho quindi diritto allo sconto del 10% sino al 31/12/90

Per ricevere la Jackson Card '90 l'ordine minimo è di L 100.000

Per ogni ordine verrà addebitato l'importo di L. 4.500 quale contributo fisso di spedizione

MODALITA' DI PAGAMENTO

❑ Contro assegno postale ❑ Assegno allegato n ______
Banca ______
❑ Versamento su c/c postale 11666203 a Voi intestato del quale allego fotocopia della ricevuta
❑ Carta di credito ❑ American Express ❑ Visa ❑ Diner Club ❑ Carta Si
Autorizzo l'organizzazione sopra indicata ad addebitare l'importo sul mio
Conto n° ______ Scadenza ______
❑ Richiedo l'emissione della fattura (formula riservata alle aziende) e comunico il numero di Partita I.V.A ______
Data ______ Firma ______

COMPILARE E SPEDIRE IN BUSTA CHIUSA A: GRUPPO EDITORIALE JACKSON - VIA ROSELLINI 12 - 20124 MILANO

CEDOLA DI COMMISSIONE LIBRARIA

La cedola di commissione libraria presenta una serie di domande a cui preghiamo vivamente di rispondere con precisione. I dati serviranno a qualificare meglio le caratteristiche dei lettori per poter offrire un servizio informativo sulle nuove pubblicazioni e altre novità Jackson adeguato alle esigenze specifiche di ciascuno.

NUMERO DI DIPENDENTI
A ☐ da 1 a 49 **C** ☐ da 250 a 999
B ☐ da 50 a 249 **D** ☐ da 1000 in su

SETTORE AZIENDALE
A ☐ Acquisti
B ☐ Vendite
C ☐ Progettazione/Ricerca e Sviluppo
D ☐ Marketing e Comunicazione
E ☐ Produzione
F ☐ Amministrazione/Personale/Finanza
G ☐ EDP
H ☐ Altro (specificare)

ATTIVITA' PRINCIPALE DELL'AZIENDA
P ☐ Produzione
D ☐ Distribuzione
S ☐ Servizi

SETTORE MERCEOLOGICO
A ☐ Informatica
B ☐ Automazione Industriale
C ☐ Meccanica
D ☐ Elettronica
E ☐ Strumentazione
F ☐ Elettricità e Energia
G ☐ Trasmissione Dati e Telecomunicazioni
H ☐ Finanza/Banche/Assicurazioni
I ☐ Editoria/Pubblicità/Comunicazione
L ☐ Pubblica Amministrazione Centrale/Locale
M ☐ Consulenza
N ☐ Istruzione (Scuola/Università)
O ☐ Altro (specificare)

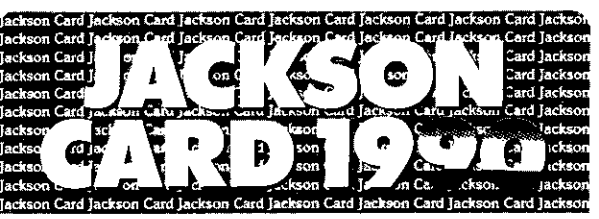
POSIZIONE
A ☐ Alta Direzione
B ☐ EDP o Technical Manager
C ☐ Dirigente
D ☐ Tecnico/Progettista
E ☐ Consulente/Professionista
F ☐ Docente/Formatore
G ☐ Studente
H ☐ Altro (specificare)

INTERESSI PRINCIPALI
01 ☐ EDP
02 ☐ Personal Computer
03 ☐ Computer Grafica e Desktop Publishing
04 ☐ Trasmissione Dati e Reti
05 ☐ Home Computer e Videogiochi
06 ☐ Automazione Industriale
07 ☐ Meccanica
08 ☐ Strumentazione Elettronica
09 ☐ Telecomunicazioni e Telefonia
10 ☐ Elettronica Professionale
11 ☐ Elettronica Hobbistica
12 ☐ Elettrotecnica e Impianti Elettrici
13 ☐ Strumenti Musicali
14 ☐ Marketing e Management
15 ☐ Broadcast/Audio e Video Professionale
16 ☐ Didattica
17 ☐ Altro (specificare)

CHE PERSONAL COMPUTER POSSIEDE/UTILIZZA
DOS ☐ MS DOS, OS/2 e PC compatibili
MAC ☐ Macintosh
AMG ☐ Amiga
C64 ☐ Commodore 64
VAR ☐ Altro Home Computer (spec.)

...Abbonati alle riviste o acquista libri Jackson per almeno L. 100.000

Avere la Jackson Card é davvero facile: basta abbonarsi oppure acquistare libri Jackson per almento L.100.000 compilando le cedole che si trovano in questa pagina.Col primo numero del '90 di Jackson Preview Magazine ogni titolare riceverà la Jackson Card 90, l'elenco dei negozi convenzionati e lo speciale buono acquisto.

GRUPPO EDITORIALE JACKSON

SERVIZIO QUALIFICAZIONE ABBONATI

ABBONAMENTO GRATUITO A 40 NUMERI, A SCELTA TRA LE SEGUENTI RIVISTE
☐ EO NEWS SETTIMANALE ☐ INFORMATICA OGGI SETTIMANALE
BARRARE LA CASELLA RELATIVA ALLA RIVISTA PRESCELTA

COGNOME E NOME ____________
VIA E NUMERO ____________
CAP ______ CITTA' ____________ PROV. ______
TEL. (______) ____________ ANNO DI NASCITA 19 ______
TITOLO DI STUDIO: ☐ LAUREA ☐ MEDIA SUPERIORE ☐ MEDIA INFERIORE
NUMERO JACKSON CARD: GOLD ____________ SILVER ____________
DITTA O ENTE ____________
VIA E NUMERO ____________
CAP ______ CITTA' ____________ PROV. ______
TEL. (______) ____________ TELEX/FAX ____________

NUMERO DI DIPENDENTI
A ☐ da 1 a 49 **C** ☐ da 250 a 999
B ☐ da 50 a 249 **D** ☐ da 1000 in su

SETTORE AZIENDALE
A ☐ Acquisti
B ☐ Vendite
C ☐ Progettazione/Ricerca e Sviluppo
D ☐ Marketing e Comunicazione
E ☐ Produzione
F ☐ Amministrazione/Personale/Finanza
G ☐ EDP
H ☐ Altro (specificare)

ATTIVITA' PRINCIPALE DELL'AZIENDA
P ☐ Produzione
D ☐ Distribuzione
S ☐ Servizi

SETTORE MERCEOLOGICO
A ☐ Informatica
B ☐ Automazione Industriale
C ☐ Meccanica
D ☐ Elettronica
E ☐ Strumentazione
F ☐ Elettricità e Energia
G ☐ Trasmissione Dati e Telecomunicazioni
H ☐ Finanza/Banche/Assicurazioni
I ☐ Editoria/Pubblicità/Comunicazione
L ☐ Pubblica Amministrazione Centrale/Locale
M ☐ Consulenza
N ☐ Istruzione (Scuola/Università)
O ☐ Altro (specificare)

POSIZIONE
A ☐ Alta Direzione
B ☐ EDP o Technical Manager
C ☐ Dirigente
D ☐ Tecnico/Progettista
E ☐ Consulente/Professionista
F ☐ Docente/Formatore
G ☐ Studente
H ☐ Altro (specificare)

INTERESSI PRINCIPALI
01 ☐ EDP
02 ☐ Personal Computer
03 ☐ Computer Grafica e Desktop Publishing
04 ☐ Trasmissione Dati e Reti
05 ☐ Home Computer e Videogiochi
06 ☐ Automazione Industriale
07 ☐ Meccanica
08 ☐ Strumentazione Elettronica
09 ☐ Telecomunicazioni e Telefonia
10 ☐ Elettronica Professionale
11 ☐ Elettronica Hobbistica
12 ☐ Elettrotecnica e Impianti Elettrici
13 ☐ Strumenti Musicali
14 ☐ Marketing e Management
15 ☐ Broadcast/Audio e Video Professionale
16 ☐ Didattica
17 ☐ Altro (specificare)

CHE PERSONAL COMPUTER POSSIEDE/UTILIZZA
DOS ☐ MS DOS, OS/2 e PC compatibili
MAC ☐ Macintosh
AMG ☐ Amiga
C64 ☐ Commodore 64
VAR ☐ Altro Home Computer (spec.)